LOW-SHOT LEARNING FOR OBJECT RECOGNITION, DETECTION, AND SEGMENTATION

A Dissertation
Presented to
The Academic Faculty

By

Amirreza Shaban

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology

August 2020

LOW-SHOT LEARNING FOR OBJECT RECOGNITION, DETECTION, AND SEGMENTATION

Proposal Committee Members:

Dr. Byron Boots, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. James Hays
School of Interactive Computing
Georgia Institute of Technology

Dr. Dhruv Batra
School of Interactive Computing
Georgia Institute of Technology

Dr. Zsolt Kira
School of Interactive Computing
Georgia Institute of Technology

Dr. Fuxin Li
School of Electrical Engineering and Computer Science
Oregon State University

Date Approved: May 8, 2020

Repeating mistakes is a hallmark of dim consciousness.

Dave Sim

To my parents Ali and Zahra.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Byron Boots for his continuous support in both research and life. Especially, I truly appreciate the research freedom and invaluable advices he provided throughout my Ph.D. study, which have cultivated me to become an independent researcher.

I would like to thank my thesis committee: Prof. James Hayes, Prof. Dhruv Batra, Prof. Zsolt Kira, and Prof. Fuxin Li for their encouragement and insightful comments that further widen my research in various perspectives. I would like to equally thank my mentors, Dr. Omid Mohareri, Dr. Hamid Reza Vaezi Joze who I was privileged to work with during internships and have provided me priceless suggestions ever since.

Tremendous thanks also go to my wonderful friends, colleagues, and collaborators; especially, Amir Rahimi, Dr. Ching-An Cheng, Shray Bansal, Alexander Lambert, Zhen Liu, Nathan Hatch, Prof. Irfan Essa, and Prof. Richard Hartley. I learned a significant amount from these collaborations and fruitful discussions. This thesis would not be possible without their contributions. I would like to thank my friend Sara for her mental support and love during my defense semester.

Last but not the least, I would like to thank my family, especially my beloved partner Safoora, my parents, and my cats Nabat and Konji. They have been my mental support, providing me unconditional love throughout good and bad times of this journey. I would not be where I am today if not for them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Deep Neural Networks are powerful at solving classification problems in computer vision. However, learning classifiers with these models requires a large amount of labeled training data, and recent approaches have struggled to adapt to new classes in a data-efficient manner. On the other hand, the human brain is capable of utilizing already known knowledge in order to learn new concepts with fewer examples and less supervision. Many meta-learning algorithms have been proposed to fill this gap but they come with their practical and theoretical limitations. We review the well-known bi-level optimization as a general framework for few-shot learning and hyperparameter optimization and discuss the practical limitations of computing the full gradient. We provide theoretical guarantees for the convergence of the bi-level optimization using the approximated gradients computed by the truncated back-propagation. In the next step, we propose an empirical method for few-shot semantic segmentation: instead of solving the inner optimization, we propose to directly estimate its result by a general function approximator. Finally, we will discuss extensions of this work with the focus on weakly-supervised object detection when full supervision is not available for the few training examples.

CHAPTER 1
INTRODUCTION AND BACKGROUND

Knowledge transfer enables the human brain to learn new concepts quickly with as few as possible training examples. When we look at a new environment, we heavily utilize our past knowledge about objects and the way they interact to learn new concepts as fast as possible. When we see a new animal we automatically pay more attention to distinctive features in animals: is it a mammal? does it have fur? what is the color? When we see a new car we first pay attention to the body style, and light and bumper designs. Utilizing the past information does not only makes it possible to learn with fewer examples but also allows learning with less supervision: While a child needs to grasp and play with a new toy for some amount of time, a grown up could have a good estimate of objects' properties by just having a quick look.

Knowledge transfer also plays an important role in machine learning. In computer vision, pre-training on ImageNet dataset is an effective method in training deep neural networks for new tasks especially when the training sample size is limited. In hand-engineered methods like pre-training, one should carefully decide what part of the information could be transferred and what part the information need to be learned from the new task and incorporate that in their architecture design. For example, it is empirically shown that early layers of a pre-trained network learn low-level information that are transferable from one task to another while the last layers usually learn high-level task-specific information that should be tuned for the task at hand. Thus, in fine-tuning a pre-trained network, the weights of the early layers are usually frozen or get updated by a lower learning rate compared to the other parts of the network. Number of frozen layers and the amount of fine-tuning have to be manually decided based on the task and the number of training examples.

In contrast to the hand-engineered transfer learning methods, there are meta-learning (Franceschi

et al., 2017a) (learning-to-learn) algorithms that use a data driven approach to learn the transferable knowledge. The goal in meta-learning is to use machine learning techniques to learn the common information among multiple tasks and utilize it to learn a new task. This naturally leads to a bi-level learning strategy: an inner learning algorithm that learns a new task given the transferred knowledge (learning biases) and cues, and an outer learning algorithm that is concerned about learning a set of useful biases that could be generalized from one task to another. Mathematically, these problems can be formulated as a stochastic optimization problem with an equality constraint:

$$\begin{aligned} \min_{\lambda} F(\lambda) &:= \mathbb{E}_S \left[f_S(\hat{w}_S^*(\lambda), \lambda) \right] \\ \text{s.t.} \quad \hat{w}_S^*(\lambda) &\approx_\lambda \arg\min_{w} g_S(w, \lambda) \end{aligned} \tag{1.1}$$

where w and λ are the *parameter* and the *hyper-parameter*, F and f_S are the expected and the sampled *upper-level objective*, g_S is the sampled *lower-level objective*, and S is a random variable called the *context*. The notation $\approx_\lambda$ means that $\hat{w}_S^*(\lambda)$ equals the unique return value of a prespecified iterative algorithm (e.g. gradient descent) that approximately finds a local minimum of g_S. This algorithm is part of the problem definition and can also be parametrized by λ (e.g. step size). The motivation for explicitly considering the approximate solution $\hat{w}_S^*(\lambda)$ rather than an exact minimizer w_S^* of g_S is that w_S^* is usually not available in closed form. This setup enables λ to account for the imperfections of the lower-level optimization algorithm.

Solving the bilevel optimization problem in (1.1) is challenging due to the complicated dependency of the upper-level problem on λ induced by $\hat{w}_S^*(\lambda)$. This difficulty is further aggravated when λ and w are high-dimensional, precluding the use of black-box optimization techniques such as grid/random search (Bergstra and Bengio, 2012) and Bayesian optimization (Snoek, Larochelle, and Adams, 2012; Srinivas et al., 2010).

In Chapter 2, we propose to use approximated gradients computed by truncated back-propagation method to optimize the bi-level optimization problem. We analyze the properties

of this family of approximate gradients and establish sufficient conditions for convergence. We validate this hyperparameter tuning, hyper-data cleaning, and few-shot image classification tasks. We find that optimization with the approximate gradient computed using few-step back-propagation often performs comparably to optimization with the exact gradient, while requiring far less memory and half the computation time.

In Chapter 3, we consider the problem of few-shot image segmentation which is more complicated than few-shot image classification. We present an alternative direction to overcome the complexity of solving the bi-level optimization for this problem. We replace the inner optimization problem with a universal function approximator. This way the function approximator is learned by optimizing the outer optimization taking standard back-propagation through the function approximator. Specifically, we train a network that, given a small set of annotated images, produces parameters for a Fully Convolutional Network (FCN). We use this FCN to perform dense pixel-level prediction on a test image for the new semantic class. Our architecture shows a 25% relative meanIoU improvement compared to the best baseline methods for one-shot segmentation on unseen classes in the PASCAL VOC 2012 dataset and is at least 3 times faster.

In Chapter 4, as an application of one-shot image segmentation, we present a novel approach to video object segmentation which reconciles unsupervised, semantic and semi-supervised video segmentation. In contrast with the state-of-the-art semi-supervised video segmentation methods that require all the objects be annotated in the first frame, our flexible framework requires annotations only for a sparse set of objects. The method has two main components: in the first component we extract video object proposals from each frame. For the objects that are not annotated in the first frame, we utilize unsupervised or instance-aware semantic segmentation algorithm(s) to generate proposals. For objects annotated in the first frame, we develop a new one-shot segmentation algorithm to generate sequence-specific proposals that match the human-annotated proposals. In the second component, we use segment proposal tracking (SPT) to generate spatio-temporal video object proposals, which

can start in any frame. We extend SPT with a semi-Markov motion model, backtracking a segment started from frame T to the 1st frame, and a "re-tracking" capability that learns a better object appearance model after inference on the full video. Even when annotating less than 20% of the target objects in the first frame, our model shows great improvement over the state-of-the-art semi-supervised algorithms that require all the objects to be annotated in the first frame.

In Chapter 5, we focus on the problem of few-shot learning with weak supervision. Our goal is to learn new object representations by seeing few images of new objects while, unlike Chapter 3, we do not assume that object annotations (segmentation mask or bounding boxes) are available. Knowing that the new object is present somewhere in each input image, we focus on localizing the common object across input images. Given a collection of bags where each bag is a set of proposals from each image, we select one image from each bag such that the selected images are from the same object class. We model the selection as an energy minimization problem with unary and pairwise potential functions. Inspired by recent few-shot learning algorithms, we propose an approach to learn the potential functions directly from the data. We utilize this method for the task of few-shot weakly supervised object detection. Our experiments show that learning the pairwise and unary terms greatly improves the performance of the model over several well-known methods for these tasks.

Although most of the proposed techniques in this thesis are applied to computer vision applications, their applicability may not limited to few-shot learning in computer vision as we only make general assumptions about input data distribution.

1.1 Author Contributions and Collaborators

Most of the algorithms proposed in this dissertation has been published or are under review in peer reviewed conferences and are result of collaboration with different institutes and people.

The work in approximating hyper gradients for bi-level optimization in Chapter 2 has been

published in International Conference on Artificial Intelligence and Statistics (AISTATS). Refer to the original manuscript (Shaban et al., 2019b) for the list of collaborators.

The work in one-shot semantic segmentation has appeared in the British Machine Vision Conference (BMVC) Shaban et al., 2017b. The video object segmentation algorithm in Chapter 4 is developed in collaboration with a team in Oregon State University led by Prof. Fuxin Li. The proposed method is presented in DAVIS Challenge on Video Object Segmentation Workshop in conjunction with CVPR (Shaban et al., 2017a). The author of this dissertation was the lead of the team in this challenge. His main contributions are in extracting instance-aware and sequence-specific object proposals using the proposed one-shot image segmentation algorithm.

The methods in Chapter 5 are developed in a close collaboration with a group led by Prof. Richard Hartley at Australian National University (ANU). The first part of the chapter on finding the common object across few image collections has been published in International Conference on Computer Vision (ICCV) (Shaban et al., 2019a). While the author of this thesis was fully involved in all parts of the ICCV paper, his main contributions are in developing the framework for learning the pairwise potentials and application of energy minimization to the co-localization problem. He also proposed the proposed few-shot weakly supervised object detection in Chapter 5.4. An extended version of the work in Chapter 5.3 has been published in Arxiv (Rahimi et al., 2020) and is currently under review in the 2020 European Conference on Computer Vision (ECCV).

CHAPTER 2

TRUNCATED BACK-PROPAGATION FOR BILEVEL OPTIMIZATION

In this chapter we focus on solving the bi-level optimization problem in Equation 1.1. Recently, first-order bilevel optimization techniques have been revisited to solve these problems. These methods rely on an estimate of the Jacobian $\nabla_\lambda \hat{w}_S^*(\lambda)$ to optimize λ. Pedregosa, 2016 and Gould et al., 2016 assume that $\hat{w}_S^*(\lambda) = w_S^*$ and compute $\nabla_\lambda \hat{w}_S^*(\lambda)$ by implicit differentiation. By contrast, Maclaurin, Duvenaud, and Adams, 2015 and Franceschi et al., 2017b treat the iterative optimization algorithm in the lower-level problem as a dynamical system, and compute $\nabla_\lambda \hat{w}_S^*(\lambda)$ by automatic differentiation through the dynamical system. In comparison, the latter approach is less sensitive to the optimality of $\hat{w}_S^*(\lambda)$ and can also learn hyperparameters that control the lower-level optimization process (e.g. step size). However, due to superlinear time or space complexity (see Section 2.2.2), neither of these methods is applicable when both λ and w are high-dimensional Franceschi et al., 2017b.

Few-step reverse-mode automatic differentiation Baydin et al., 2018; Luketina et al., 2016 and few-step forward-mode automatic differentiation Franceschi et al., 2017b have recently been proposed as heuristics to address this issue. By ignoring long-term dependencies, the time and space complexities to compute approximate gradients can be greatly reduced. While exciting empirical results have been reported, the theoretical properties of these methods remain unclear.

In this chapter, we study the theoretical properties of these *truncated back-propagation* approaches. We show that, when the lower-level problem is locally strongly convex around $\hat{w}_S^*(\lambda)$, on-average convergence to an ϵ-approximate stationary point is guaranteed by $O(\log 1/\epsilon)$-step truncated back-propagation. We also identify additional problem structures for which asymptotic convergence to an exact stationary point is guaranteed. Empirically,

we verify the utility of this strategy for hyperparameter optimization and meta learning tasks. We find that, compared to optimization with full back-propagation, optimization with truncated back-propagation usually shows competitive performance while requiring half as much computation time and significantly less memory.

2.1 Applications

Hyperparameter Optimization The goal of hyperparameter optimization Bengio, 2000; Larsen et al., 1996 is to find hyperparameters λ for an optimization problem P such that the approximate solution $\hat{w}^*(\lambda)$ of P has low cost $c(\hat{w}^*(\lambda))$ for some cost function c. In general, λ can parametrize both the objective of P and the algorithm used to solve P. This setup is a special case of the bilevel optimization problem (1.1) where the upper-level objective c does not depend directly on λ. In contrast to meta learning (discussed below), c can be deterministic Franceschi et al., 2017b. See Section 2.4.2 for examples.

Many low-dimensional problems, such as choosing the learning rate and regularization constant for training neural networks, can be effectively solved with grid search. However, problems with thousands of hyperparameters are increasingly common, for which gradient-based methods are more appropriate Chen, Ranftl, and Pock, 2014; Maclaurin, Duvenaud, and Adams, 2015.

Meta Learning Another important application of bilevel optimization, meta learning (or learning-to-learn) uses statistical learning to optimize an algorithm $\mathcal{A}_\lambda$ over a distribution of tasks $\mathcal{T}$ and contexts S:

$$\min_\lambda \mathbb{E}_\mathcal{T} \mathbb{E}_{S|\mathcal{T}} \left[c_\mathcal{T}(\mathcal{A}_\lambda(S)) \right]. \tag{2.1}$$

It treats $\mathcal{A}_\lambda$ as a parametric function, with hyperparameter λ, that takes task-specific context information S as input and outputs a decision $\mathcal{A}_\lambda(S)$. The goal of meta learning is to optimize the algorithm's performance $c_\mathcal{T}$ (e.g. the generalization error) across tasks $\mathcal{T}$

through empirical observations. This general setup subsumes multiple problems commonly encountered in the machine learning literature, such as multi-task learning Caruana, 1998; Ranjan, Patel, and Chellappa, 2017 and few-shot learning Fei-Fei, Fergus, and Perona, 2006; Ravi and Larochelle, 2017a; Snell, Swersky, and Zemel, 2017.

Bilevel optimization emerges from meta learning when the algorithm computes $\mathcal{A}_\lambda(S)$ by internally solving a *lower-level* minimization problem with variable w. The motivation to use this class of algorithms is that the lower-level problem can be designed so that, even for tasks $\mathcal{T}$ distant from the training set, $\mathcal{A}_\lambda$ falls back upon a sensible optimization-based approach Baydin et al., 2018; Finn, Abbeel, and Levine, 2017a. By contrast, treating $\mathcal{A}_\lambda$ as a general function approximator relies on the availability of a large amount of meta training data Andrychowicz et al., 2016; Li and Malik, 2017.

In other words, the decision is $\mathcal{A}_\lambda(S) = (\hat{w}_S^*(\lambda), \lambda)$ where $\hat{w}_S^*(\lambda)$ is an approximate minimizer of some function $g_S(w, \lambda)$. Therefore, we can identify

$$\mathbb{E}_{\mathcal{T}|S}\left[c_{\mathcal{T}}(\hat{w}_S^*(\lambda), \lambda)\right] =: f_S(\hat{w}_S^*(\lambda), \lambda) \tag{2.2}$$

and write (2.1) as (1.1).[1] Compared with λ, the lower-level variable w is usually task-specific and fine-tuned based on the given context S. For example, in few-shot learning, a warm start initialization or regularization function (λ) can be learned through meta learning, so that a task-specific network (w) can be quickly trained using regularized empirical risk minimization with few examples S. See Section 2.4.3 for an example.

2.2 Bilevel Optimization

2.2.1 Setup

Let $\lambda \in \mathbb{R}^N$ and $w \in \mathbb{R}^M$. We consider solving (1.1) with first-order methods that sample S (like stochastic gradient descent) and focus on the problem of computing the gradients

[1] We have replaced $\mathbb{E}_{\mathcal{T}}\mathbb{E}_{S|\mathcal{T}}$ with $\mathbb{E}_S\mathbb{E}_{\mathcal{T}|S}$, which is valid since both describe the expectation over the joint distribution. The algorithm $\mathcal{A}_\lambda$ only perceives S, not $\mathcal{T}$.

for a given S. Therefore, we will simplify the notation below by omitting the dependency of variables and functions on S and λ (e.g. we write $\hat{w}^*_S(\lambda)$ as $\hat{w}^*$ and g_S as g). We use d_x to denote the total derivative with respect to a variable x, and ∇_x to denote the partial derivative, with the convention that $\nabla_\lambda f \in \mathbb{R}^N$ and $\nabla_\lambda \hat{w}^* \in \mathbb{R}^{N\times M}$.

To optimize λ, stochastic first-order methods use estimates of the gradient $\mathrm{d}_\lambda f = \nabla_\lambda f + \nabla_\lambda \hat{w}^* \nabla_{\hat{w}^*} f$. Here we assume that both $\nabla_\lambda f \in \mathbb{R}^N$ and $\nabla_{\hat{w}^*} f \in \mathbb{R}^M$ are available through a stochastic first-order oracle, and focus on the problem of computing the matrix-vector product $\nabla_\lambda \hat{w}^* \nabla_{\hat{w}^*} f$ when both λ and w are high-dimensional.

2.2.2 Computing the hypergradient

Like Franceschi et al., 2017b; Maclaurin, Duvenaud, and Adams, 2015, we treat the iterative optimization algorithm that solves the lower-level problem as a dynamical system. Given an initial condition $w_0 = \Xi_0(\lambda)$ at $t = 0$, the update rule can be written as[2]

$$w_{t+1} = \Xi_{t+1}(w_t, \lambda), \qquad \hat{w}^* = w_T \tag{2.3}$$

in which Ξ_t defines the transition and and T is the number iterations performed. For example, in gradient descent, $\Xi_{t+1}(w_t, \lambda) = w_t - \gamma_t(\lambda)\nabla_w g(w_t, \lambda)$, where $\gamma_t(\lambda)$ is the step size.

By unrolling the iterative update scheme (2.3) as a computational graph, we can view $\hat{w}^*$ as a function of λ and compute the required derivative $\mathrm{d}_\lambda f$ Baydin et al., 2017. Specifically, it can be shown by the chain rule[3]

$$\mathrm{d}_\lambda f = \nabla_\lambda f + \textstyle\sum_{t=0}^{T} B_t A_{t+1} \cdots A_T \nabla_{\hat{w}^*} f \tag{2.4}$$

where $A_{t+1} = \nabla_{w_t} \Xi_{t+1}(w_t, \lambda)$, $B_{t+1} = \nabla_\lambda \Xi_{t+1}(w_t, \lambda)$ for $t \geq 0$, and $B_0 = \mathrm{d}_\lambda \Xi_0(\lambda)$.

The computation of (2.4) can be implemented either in reverse mode or forward

[2]For notational simplicity, we consider the case where w_t is the state of (2.3); our derivation can be easily generalized to include other internal states, e.g. momentum.

[3]Note that this assumes g is twice differentiable.

Table 2.1: Comparison of the additional time and space to compute $\mathrm{d}_\lambda f = \nabla_\lambda f + \nabla_\lambda \hat{w}^* \nabla_{\hat{w}^*} f$, where $\lambda \in \mathbb{R}^N$, $w \in \mathbb{R}^M$, and $c = c(M, N)$ is the time complexity to compute the transition function Ξ. †Checkpointing doubles the constant in time complexity, compared with other approaches.

METHOD	TIME	SPACE	EXACT
FMD	$O(cNT)$	$O(MN)$	✓
RMD	$O(cT)$	$O(MT)$	✓
CHECKPOINTING EVERY $\sqrt{T}$ STEPS†	$O(cT^\dagger)$	$O(M\sqrt{T})$	✓
K-RMD	$O(cK)$	$O(MK)$	

mode Franceschi et al., 2017b. Reverse-mode differentiation (RMD) computes (2.4) by back-propagation:

$$\begin{aligned} \alpha_T &= \nabla_{\hat{w}^*} f, \quad h_T = \nabla_\lambda f, \\ h_{t-1} &= h_t + B_t \alpha_t, \quad \alpha_{t-1} = A_t \alpha_t \end{aligned} \tag{2.5}$$

and finally $\mathrm{d}_\lambda f = h_{-1}$. Forward-mode differentiation (FMD) computes (2.4) by forward propagation:

$$\begin{aligned} Z_0 &= B_0, \quad Z_{t+1} = Z_t A_{t+1} + B_{t+1}, \\ \mathrm{d}_\lambda f &= Z_T \nabla_{\hat{w}^*} f + \nabla_\lambda f \end{aligned} \tag{2.6}$$

The choice between RMD and FMD is a trade-off based on the size of $w \in \mathbb{R}^M$ and $\lambda \in \mathbb{R}^N$ (see Table 2.1 for a comparison). For example, one drawback of RMD is that all the intermediate variables $\{w_t \in \mathbb{R}^M\}_{t=1}^T$ need to be stored in memory in order to compute A_t and B_t in the backward pass. Therefore, RMD is only applicable when MT is small, as in Finn, Abbeel, and Levine, 2017a. Checkpointing Hascoet and Araya-Polo, 2006 can reduce this to $M\sqrt{T}$, but it *doubles* the computation time. Complementary to RMD, FMD propagates the matrix $Z_t \in \mathbb{R}^{M \times N}$ in line with the forward evaluation of the dynamical system (2.3), and does not require any additional memory to save the intermediate variables.

However, propagating the matrix Z_t instead of vectors requires memory of size MN and is N-times slower compared with RMD.

2.3 Truncated Back-propagation

In this chapter, we investigate approximating (2.4) with partial sums, which was previously proposed as a heuristic for bilevel optimization (Luketina et al., 2016 Eq. 3, Baydin et al., 2018 Eq. 2). Formally, we perform K-step truncated back-propagation (K-RMD) and use the intermediate variable h_{T-K} to construct an approximate gradient:

$$h_{T-K} = \nabla_\lambda f + \sum_{t=T-K+1}^{T} B_t A_{t+1} \cdots A_T \nabla_{\hat{w}^*} f \tag{2.7}$$

This approach requires storing only the last K iterates w_t, and it also saves computation time. Note that K-RMD can be combined with checkpointing for further savings, although we do not investigate this.

2.3.1 General properties

We first establish some intuitions about why using K-RMD to optimize λ is reasonable. While building up an approximate gradient by truncating back-propagation in general optimization problems can lead to large bias, the bilevel optimization problem in (1.1) has some nice structure. Here we show that if the lower-level objective g is locally strongly convex around $\hat{w}^*$, then the bias of h_{T-K} can be exponentially small in K. That is, choosing a small K would suffice to give a good gradient approximation in finite precision. The proof is given in Appendix A.1.

Proposition 2.3.1. *Assume g is β-smooth, twice differentiable, and locally α-strongly convex in w around $\{w_{T-K-1}, \ldots, w_T\}$. Let $\Xi_{t+1}(w_t, \lambda) = w_t - \gamma \nabla_w g(w_t, \lambda)$. For $\gamma \leq \frac{1}{\beta}$, it holds*

$$\|h_{T-K} - \mathrm{d}_\lambda f\| \leq 2^{T-K+1}(1-\gamma\alpha)^K \|\nabla_{\hat{w}^*} f\| M_B \tag{2.8}$$

where $M_B = \max_{t\in\{0,\dots,T-K\}} \|B_t\|$. *In particular, if* g *is globally* α*-strongly convex, then*

$$\|h_{T-K} - \mathrm{d}_\lambda f\| \leq \tfrac{(1-\gamma\alpha)^K}{\gamma\alpha}\|\nabla_{\hat{w}^*} f\| M_B. \tag{2.9}$$

Note $0 \leq (1-\gamma\alpha) < 1$ since $\gamma \leq \frac{1}{\beta} \leq \frac{1}{\alpha}$. Therefore, Proposition 2.3.1 says that if $\hat{w}^*$ converges to the *neighborhood* of a strict local minimum of the lower-level optimization, then the bias of using the approximate gradient of K-RMD decays exponentially in K. This exponentially decaying property is the main reason why using h_{T-K} to update the hyperparameter λ works.

Next we show that, when the lower-level problem g is second-order continuously differentiable, $-h_{T-K}$ actually is a sufficient descent direction. This is a much stronger property than the small bias shown in Proposition 2.3.1, and it is critical in order to prove convergence to exact stationary points (cf. Theorem 2.3.4). To build intuition, here we consider a simpler problem where g is globally strongly convex and $\nabla_\lambda f = 0$. These assumptions will be relaxed in the next subsection.

Lemma 2.3.2. *Let* g *be globally strongly convex and* $\nabla_\lambda f = 0$. *Assume* g *is second-order continuously differentiable and* B_t *has full column rank for all* t. *Let* $\Xi_{t+1}(w_t, \lambda) = w_t - \gamma\nabla_w g(w_t, \lambda)$. *For all* $K \geq 1$, *with* T *large enough and* γ *small enough, there exists* $c > 0$, *s.t.* $h_{T-K}^\top \mathrm{d}_\lambda f \geq c\|\nabla_{\hat{w}^*} f\|^2$. *This implies* h_{T-K} *is a sufficient descent direction, i.e.* $h_{T-K}^\top \mathrm{d}_\lambda f \geq \Omega(\|\mathrm{d}_\lambda f\|^2)$.

The full proof of this non-trivial result is given in Appendix A.2. Here we provide some ideas about why it is true. First, by Proposition 2.3.1, we know the bias decays exponentially. However, this alone is not sufficient to show that $-h_{T-K}$ is a sufficient descent direction. To show the desired result, Lemma 2.3.2 relies on the assumption that g is second-order continuously differentiable and the fact that using gradient descent to optimize a well-conditioned function has linear convergence Hazan, 2016. These two new structural properties further reduce the bias in Proposition 2.3.1 and lead to Lemma 2.3.2. Here the

full rank assumption for B_t is made to simplify the proof. We conjecture that this condition can be relaxed when $K > 1$. We leave this to future work.

2.3.2 Convergence

With these insights, we analyze the convergence of bilevel optimization with truncated back-propagation. Using Proposition 2.3.1, we can immediately deduce that optimizing λ with h_{T-K} converges on-average to an ϵ-approximate stationary point. Let $\nabla F(\lambda_\tau)$ denote the hypergradient in the τth iteration.

Theorem 2.3.3. *Suppose F is smooth and bounded below, and suppose there is $\epsilon < \infty$ such that $\|h_{T-K} - \mathrm{d}_\lambda f\| \leq \epsilon$. Using h_{T-K} as a stochastic first-order oracle with a decaying step size $\eta_\tau = O(1/\sqrt{\tau})$ to update λ with gradient descent, it follows after R iterations,*

$$\mathbb{E}\left[\sum_{\tau=1}^{R} \frac{\eta_\tau \|\nabla F(\lambda_\tau)\|^2}{\sum_{\tau=1}^{R} \eta_\tau}\right] \leq \widetilde{O}\left(\epsilon + \frac{\epsilon^2 + 1}{\sqrt{R}}\right).$$

That is, under the assumptions in Proposition 2.3.1, learning with h_{T-K} converges to an ϵ-approximate stationary point, where $\epsilon = O((1 - \gamma\alpha)^{-K})$.

We see that the bias becomes small as K increases. As a result, it is sufficient to perform K-step truncated back-propagation with $K = O(\log 1/\epsilon)$ to update λ.

Next, using Lemma 2.3.2, we show that the bias term in Theorem 2.3.3 can be removed if the problem is more structured. As promised, we relax the simplifications made in Lemma 2.3.2 into assumptions 2 and 3 below and only assume g is locally strongly convex.

Theorem 2.3.4. *Under the assumptions in Proposition 2.3.1 and Theorem 2.3.3, if in addition*

1. *g is second-order continuously differentiable*
2. *B_t has full column rank around w_T*
3. *$\nabla_\lambda f^\top (\mathrm{d}_\lambda f + h_{T-K} - \nabla_\lambda f) \geq \Omega(\|\nabla_\lambda f\|^2)$*

4. the problem is deterministic (i.e. $F = f$)

then for all $K \geq 1$, with T large enough and γ small enough, the limit point is an exact stationary point, i.e. $\lim_{\tau \to \infty} \|\nabla F(\lambda_\tau)\| = 0$.

Theorem 2.3.4 shows that if the partial derivative $\nabla_\lambda f$ does not interfere strongly with the partial derivative computed through back-propagating the lower-level optimization procedure (assumption 3), then optimizing λ with h_{T-K} converges to an *exact* stationary point. This is a very strong result for an interesting special case. It shows that even with one-step back-propagation h_{T-1}, updating λ can converge to a stationary point.

This non-interference assumption unfortunately is necessary; otherwise, truncating the full RMD leads to constant bias, as we show below (proved in Appendix A.5).

Theorem 2.3.5. *There is a problem, satisfying all but assumption 3 in Theorem 2.3.4, such that optimizing λ with h_{T-K} does not converge to a stationary point.*

Note however that the non-interference assumption is satisfied when $\nabla_\lambda f = 0$, i.e. when the upper-level problem does not directly depend on the hyperparameter. This is the case for many practical applications: e.g. hyperparameter optimization, meta-learning regularization models, image desnosing Chen, Ranftl, and Pock, 2014; Roth and Black, 2005, data hyper-cleaning Franceschi et al., 2017b, and task interaction Evgeniou, Micchelli, and Pontil, 2005.

2.3.3 Relationship with implicit differentiation

The gradient estimate h_{T-K} is related to implicit differentiation, which is a classical first-order approach to solving bilevel optimization problems Bengio, 2000; Larsen et al., 1996. Assume g is second-order continuously differentiable and that its optimal solution uniquely exists such that $w^* = w^*(\lambda)$. By the implicit function theorem Rudin, 1964, the total

derivative of f with respect to λ can be written as

$$\mathrm{d}_\lambda f = \nabla_\lambda f - \nabla_{\lambda,w} g \nabla_{w,w}^{-1} g \nabla_{\hat{w}^*} f \tag{2.10}$$

where all derivatives are evaluated at $(w^*(\lambda), \lambda)$ and $\nabla_{\lambda,w} g = \nabla_\lambda(\nabla_w g) \in \mathbb{R}^{N \times M}$.

Here we show that, in the limit where $\hat{w}^*$ converges to w^*, h_{T-K} can be viewed as approximating the matrix inverse in (2.10) with an order-K Taylor series. This can be seen from the next proposition.

Proposition 2.3.6. *Under the assumptions in Proposition 2.3.1, suppose w_t converges to a stationary point w^*. Let $A_\infty = \lim_{t\to\infty} A_t$ and $B_\infty = \lim_{t\to\infty} B_t$. For $\gamma < \frac{1}{\beta}$, it satisfies that*

$$-\nabla_{\lambda,w} g \nabla_{w,w}^{-1} g = B_\infty \textstyle\sum_{k=0}^{\infty} A_\infty^k \tag{2.11}$$

By Proposition 2.3.6, we can write $\mathrm{d}_\lambda f$ in (2.10) as

$$\begin{aligned} \mathrm{d}_\lambda f &= \nabla_\lambda f - \nabla_{\lambda,w} g \nabla_{w,w}^{-1} g \nabla_{\hat{w}^*} f \\ &= h_{T-K} + B_\infty \sum_{k=K}^{\infty} A_\infty^k \nabla_{\hat{w}^*} f \end{aligned}$$

That is, h_{T-K} captures the first K terms in the Taylor series, and the residue term has an upper bound as in Proposition 2.3.1.

Given this connection, we can compare the use of h_{T-K} and approximating (2.10) using K steps of conjugate gradient descent for high-dimensional problems Pedregosa, 2016. First, both approaches require local strong-convexity to ensure a good approximation. Specifically, let $\kappa = \frac{\beta}{\alpha} > 0$ locally around the limit. Using h_{T-K} has a bias in $O((1 - \frac{1}{\kappa})^K)$, whereas using (2.10) and inverting the matrix with K iterations of conjugate gradient has a bias in $O((1 - \frac{1}{\sqrt{\kappa}})^K)$ Shewchuk, 1994. Therefore, when w^* is available, solving (2.10) with conjugate gradient descent is preferable. However, in practice, this is hardly true. When an

approximate solution $\hat{w}^*$ to the lower-level problem is used, adopting (2.10) has no control on the approximate error, nor does it necessarily yield a descent direction. On the contrary, h_{T-K} is based on Proposition 2.3.1, which uses a weaker assumption and does not require the convergence of w_t to a stationary point. Truncated back-propagation can also optimize the hyperparameters that control the lower-level optimization process, which the implicit differentiation approach cannot do.

2.4 Experiments

2.4.1 Toy problem

Consider the following simple problem for $\lambda, w \in \mathbb{R}^2$:

$$\min_{\lambda} \|\hat{w}^*\|^2 + 10\|\sin(\hat{w}^*)\|^2 =: f(\hat{w}^*, \lambda)$$
$$\text{s.t. } \hat{w}^* \approx \arg\min_{w} \tfrac{1}{2}(w-\lambda)^\top G(w-\lambda) =: g(w, \lambda)$$

where $\|\cdot\|$ is the ℓ_2 norm, sine is applied elementwise, $G = \mathrm{diag}(1, \frac{1}{2})$, and we define $\hat{w}^*$ as the result of $T = 100$ steps of gradient descent on g with learning rate $\gamma = 0.1$, initialized at $w_0 = (2, 2)$. A plot of $f(\cdot, \lambda)$ is shown in Figure. 2.1. We will use this problem to visualize the theorems and explore the empirical properties of truncated back-propagation.

This deterministic problem satisfies all of the assumptions in the previous section, particularly those of Theorem 2.3.4: g is 1-smooth and $\frac{1}{2}$-strongly convex, with

$$B_{t+1} = \nabla_\lambda[w_t - \gamma \nabla_w g(w_t, \lambda)] = \gamma G$$

and $B_0 = 0$. Although f is somewhat complicated, with many saddle points, it satisfies the non-interference assumption because $\nabla_\lambda f = 0$.

Figure 2.1 visualizes Proposition 2.3.1 by plotting the approximation error $\|h_{T-K} - \mathrm{d}_\lambda f\|$ and the theoretical bound $\frac{(1-\gamma\alpha)^K}{\gamma\alpha}\|\nabla_{\hat{w}^*} f\| M_B$ at $\lambda = (1, 1)$. For this problem, $\alpha = \frac{1}{2}$,

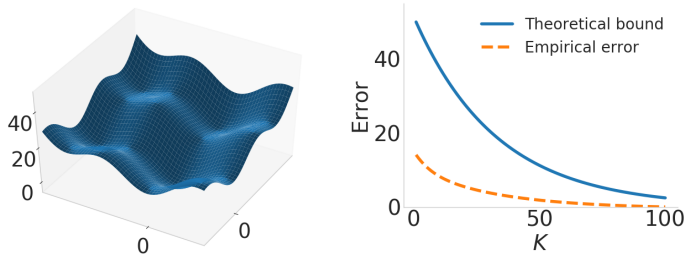


Figure 2.1: Graph of f and visualization of Prop. 2.3.1.

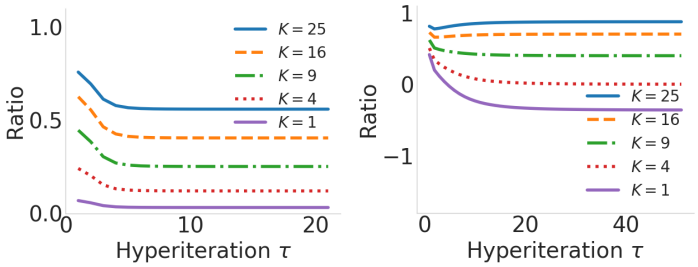


Figure 2.2: The ratio $h_{T-K}^\top \mathrm{d}_\lambda f / \|\mathrm{d}_\lambda f\|^2$ at various λ_τ, for f and $\widetilde{f}$ respectively.

$M_B = \|\gamma G\| = \gamma$, and $\nabla_{\hat{w}^*} f$ can be found analytically from $\hat{w}^* = Cw_0 + (I - C)\lambda$, where $C = (I - \gamma G)^T$. Figure 2.4 (left) plots the iterates λ_τ when optimizing f using 1-RMD and a decaying meta-learning rate $\eta_\tau = \frac{\eta_0}{\sqrt{\tau}}$.[4] In comparison with the true gradient $\mathrm{d}_\lambda f$ at these points, we see that h_{T-1} is indeed a descent direction. Figure 2.2 (left) visualizes this in a different way, by plotting $h_{T-K}^\top \mathrm{d}_\lambda f / \|\mathrm{d}_\lambda f\|^2$ for various K at each point λ_τ along the $K = 1$ trajectory. By Lemma 2.3.2, this ratio stays well away from zero.

To demonstrate the biased convergence of Theorem 2.3.3, we break assumption 3 of Theorem 2.3.4 by changing the upper objective to $\widetilde{f}(\hat{w}^*, \lambda) := f(\hat{w}^*, \lambda) + 5\|\lambda - (1, 0)\|^2$ so that $\nabla_\lambda \widetilde{f} \neq 0$. The guarantee of Lemma 2.3.2 no longer applies, and we see in Figure 2.2 (right) that $h_{T-K}^\top \mathrm{d}_\lambda f / \|\mathrm{d}_\lambda f\|^2$ can become negative. Indeed, Figure 2.3 shows that optimizing $\widetilde{f}$ with h_{T-1} converges to a suboptimal point. However, it also shows that using larger K rapidly decreases the bias.

For the original objective f, Theorem 2.3.4 guarantees exact convergence. Figure 2.4 shows optimization trajectories for various K, and a log-scale plot of their convergence rates.

[4]Because $\|h_{T-K}\|$ varies widely with K, we tune η_0 to ensure that the first update $\eta_1 h_{T-K}(\lambda_1)$ has norm 0.6.

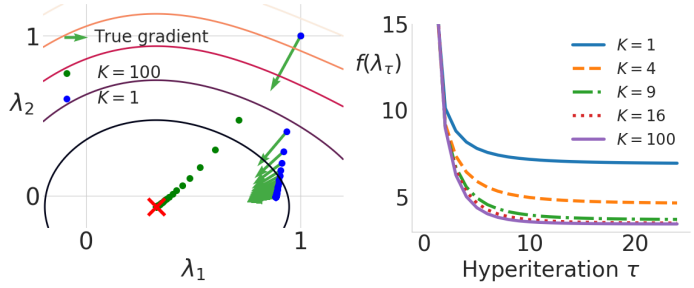


Figure 2.3: Biased convergence for $\widetilde{f}$. The red X marks the optimal λ.

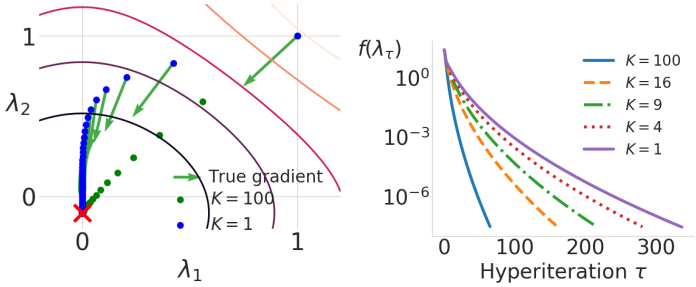


Figure 2.4: Convergence for f.

Note that, because the lower-level problem cannot be perfectly solved within T steps, the optimal λ is offset from the origin. Truncated back-propagation can handle this, but it breaks the assumptions required by the implicit differentiation approach to bilevel optimization.

2.4.2 Hyperparameter optimization problems

Data hypercleaning

In this section, we evaluate K-RMD on a hyperparameter optimization problem. The goal of data hypercleaning Franceschi et al., 2017b is to train a linear classifier for MNIST LeCun et al., 1998, with the complication that half of our training labels have been corrupted. To do this with hyperparameter optimization, let $W \in \mathbb{R}^{10\times 785}$ be the weights of the classifier, with the outer objective f measuring the cross-entropy loss on a cleanly labeled validation set. The inner objective is defined as *weighted* cross-entropy training loss plus regularization:

$$g(W, \lambda) = \sum_{i=1}^{5000} -\sigma(\lambda_i) \log(e_{y_i}^\top W x_i) + 0.001\|W\|_F^2$$

where (x_i, y_i) are the training examples, σ denotes the sigmoid function, $\lambda_i \in \mathbb{R}$, and $\|\cdot\|_F$ is the Frobenius norm. We optimize λ to minimize validation loss, presumably by decreasing the weight of the corrupted examples. The optimization dimensions are $|\lambda| = 5000$, $|W| = 7850$. Franceschi et al., 2017b previously solved this problem with full RMD, and it happens to satisfy many of our theoretical assumptions, making it an interesting case for empirical study.[5]

We optimize the lower-level problem g through $T = 100$ steps of gradient descent with $\gamma = 1$ and consider how adjusting K changes the performance of K-RMD.[6] Our hypothesis is that K-RMD for small K works almost as well as full RMD in terms of validation and test accuracy, while requiring less time and far less memory. We also hypothesize that K-RMD does almost as well as full RMD in identifying which samples were corrupted Franceschi et al., 2017b. Because our formulation of the problem is unconstrained, the weights $\sigma(\lambda_i)$ are never exactly zero. However, we can calculate an F1 score by setting a threshold on λ: if $\sigma(\lambda_i) < \sigma(-3) \approx 0.047$, then the hyper-cleaner has marked example i as corrupted.[7]

Table 2.2 reports these metrics for various K. We see that 1-RMD is somewhat worse than the others, and that validation loss (the outer objective f) decreases with K more quickly than generalization error. The F1 score is already maximized at $K = 5$. These preliminary results indicate that in situations with limited memory, K-RMD for small K (e.g. $K = 5$) may be a reasonable fallback: it achieves results close to full backprop, and it runs about twice as fast.

From a theoretical optimization perspective, we wonder whether K-RMD converges to a stationary point of f. Data hypercleaning satisfies all of the assumptions of Theorem 2.3.4 except that B_t is not full column rank (since $M < N$). In particular, the validation loss f is deterministic and satisfies $\nabla_\lambda f = 0$. Figure 2.5 plots the norm of the true gradient $\mathrm{d}_\lambda f$

[5] We have reformulated the constrained problem from Franceschi et al., 2017b as an unconstrained one that more closely matches our theoretical assumptions. For the same reason, we regularized g to make it strongly convex. Finally, we do not retrain on the hypercleaned training + validation data. This is because, for our purposes, comparing the performance of $\hat{w}^*$ across K is sufficient.

[6] See Appendix A.7.1 for more experimental setup.

[7] F1 scores for other choices of the threshold were very similar. See Appendix A.7.1 for details.

Table 2.2: Hypercleaning metrics after 1000 hyperiters.

K	Test Acc.	Val. Acc.	Val. Loss	F1
1	87.50	89.32	0.413	0.85
5	88.05	89.90	0.383	0.89
25	88.12	89.94	0.382	0.89
50	88.17	90.18	0.381	0.89
100	88.33	90.24	0.380	0.88

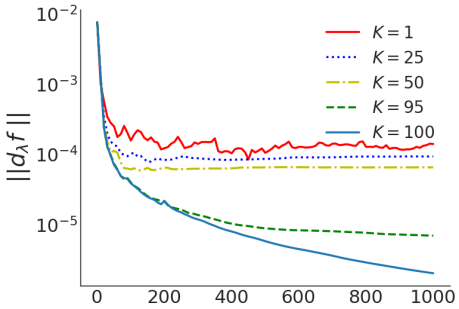


Figure 2.5: $\|\mathrm{d}_\lambda f\|$ vs. hyperiteration for hypercleaning.

on a log scale at the K-RMD iterates for various K. We see that, despite satisfying almost all assumptions, this problem exhibits biased convergence. The limit of $\|\mathrm{d}_\lambda f\|$ decreases slowly with K, but recall from Table 2.2 that practical metrics improve more quickly.

Task interaction

We next consider the problem of multitask learning Evgeniou, Micchelli, and Pontil, 2005. Similar to Franceschi et al., 2017b, we formulate this as a hyperparameter optimization problem as follows. The lower-level objective $g(w, \{C, \rho\})$ learns V different linear models with parameter set $w = \{w_v\}_{v=1}^V$:

$$l(w) + \sum_{1 \le i,j \le K} C_{ij} \|w_i - w_j\|^2 + \rho \sum_{v=1}^{V} \|w_v\|^2$$

where $l(w)$ is the training loss of the multi-class linear logistic regression model, ρ is a regularization constant, and C is a nonnegative, symmetric hyperparameter matrix that encodes the similarity between each pair of tasks. After 100 iterations of gradient descent

with learning rate 0.1, this yields $\hat{w}^*$. The upper-level objective $c(\hat{w}^*)$ estimates the linear regression loss of the learned model $\hat{w}^*$ on a validation set. Presumably, this will be improved by tuning C to reflect the true similarities between the tasks. The tasks that we consider are image recognition trained on very small subsets of the datasets CIFAR-10 and CIFAR-100.[8]

From an optimization standpoint, we are most interested in the upper-level loss on the validation set, since that is what is directly optimized, and its value is a good indication of the performance of the inexact gradient. Figure 2.6 plots this learning curve along with two other metrics of theoretical interest: norm of the true gradient, and cosine similarity between the true and approximate gradients. In CIFAR100, the validation error and gradient norm plots show that K-RMD converges to an approximate stationary point with a bias that rapidly decreases as K increases, agreeing with Proposition 2.3.1. Also, we find that negative values exist in the cosine similarity of 1-RMD, which implies that not all the assumptions in Theorem 2.3.4 hold for this problem (e.g. B_t might not be full rank, or the the inner problem might not be locally strong convex around $\hat{w}^*$.) In CIFAR10, some unusal behavior happens. For $K > 1$, the truncated gradient and the full gradient directions eventually become almost the same. We believe this is a very interesting observation but beyond the scope of the proposal to explain.

In Table 2.3, we report the testing accuracy over 10 trials. While in general increasing the number of back-propagation steps improves accuracy, the gaps are small. A thorough investigation of the relationship between convergence and generalization is an interesting open question of both theoretical and practical importance.

2.4.3 Meta-learning: One-shot classification

The aim of this experiment is to evaluate the performance of truncated back-propagation in multi-task, stochastic optimization problems. We consider in particular the one-shot classification problem Finn, Abbeel, and Levine, 2017a, where each task $\mathcal{T}$ is a k-way

[8] See Appendix A.7.2 for more details.

Table 2.3: Test accuracy for task interaction. Few-step K-RMD achieves similar performance as full RMD.

	Method	Avg. Acc.	Avg. Iter.	Sec/iter.
CIFAR-10	1-RMD	61.11 ± 1.23	3300	0.8
	5-RMD	61.33 ± 1.08	4950	1.3
	25-RMD	61.31 ± 1.24	4825	1.4
	Full RMD	61.28 ± 1.21	4500	2.2
CIFAR-100	1-RMD	34.37 ± 0.63	7440	1.0
	5-RMD	34.34 ± 0.68	8805	1.4
	25-RMD	34.51 ± 0.69	8660	1.6
	Full RMD	34.70 ± 0.64	5670	2.8

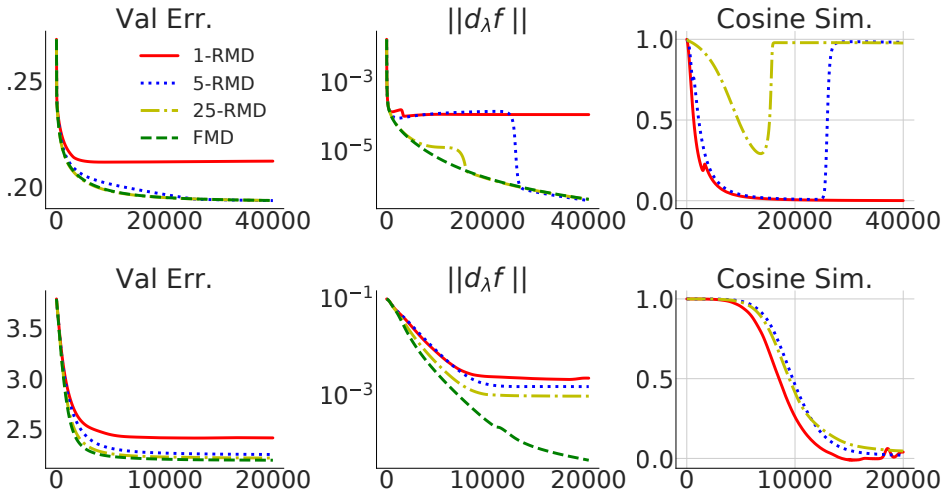


Figure 2.6: Upper-level objective loss (first column), norm of the exact gradient (second column), and cosine similarity (last column) vs. hyper-iteration on CIFAR10 (first row) and CIFAR100 (second row) datasets.

classification problem and the goal is learn a hyperparameter λ such that each task can be solved with few training samples.

In each hyper-iteration, we sample a task, a training set, and a validation set as follows: First, k classes are randomly chosen from a pool of classes to define the sampled task $\mathcal{T}$. Then the training set $S = \{(x_i, y_i)\}_{i=1}^{k}$ is created by randomly drawing one training example (x_i, y_i) from each of the k classes. The validation set Q is constructed similarly, but with more examples from each class. The lower-level objective $g_S(w, \lambda)$ is

$$\sum_{(x_i,y_i)\in S} l(nn(x_i; w, \lambda), y_i) + \sum_{j=1}^{V} \rho_j ||w_j - c_j||^2$$

where $l(\cdot, \cdot)$ is the k-way cross-entropy loss, and $nn(\cdot; w, \lambda)$ is a deep neural network parametrized by $w = \{w_1, \ldots, w_V\}$ and optionally hyperparameter λ. To prevent overfitting in the lower-level optimization, we regularize each parameter w_j to be close to center c_j with weight $\rho_j > 0$. Both c_j and ρ_j are hyperparameters, as well as the inner learning rate γ. The upper-level objective is the loss of the trained network on the sampled validation set Q. In contrast to other experiments, this is a stochastic optimization problem. Also, $\mathcal{A}_\lambda(S)(x_i) = nn(x_i; \hat{w}^*, \lambda)$ depends directly on the hyperparameter λ, in addition to the indirect dependence through $\hat{w}^*$ (i.e. $\nabla_\lambda f \neq 0$).

We use the Omniglot dataset Lake, Salakhutdinov, and Tenenbaum, 2015 and a similar neural network as used in Finn, Abbeel, and Levine, 2017a with small modifications. Please refer to Appendix A.7.3 for more details about the model and the data splits. We set $T = 50$ and optimize over the hyperparameter $\lambda = \{\lambda_{l_1}, \lambda_{l_2}, c, \rho, \gamma\}$. The average accuracy of each model is evaluated over 120 randomly sampled training and validation sets from the meta-testing dataset. For comparison, we also try using full RMD with a very short horizon $T = 1$, which is common in recent work on few-shot learning Finn, Abbeel, and Levine, 2017a.

The statistics are shown in Table 2.4 and the learning curves in Figure 2.7. In addition to saving memory, all truncated methods are faster than full RMD, sometimes even five times faster. These results suggest that running few-step back-propagation with more hyper-iterations can be more efficient than the full RMD. To support this hypothesis, we also ran 1-RMD and 10-RMD for an especially large number of hyper-iterations (15k). Even with this many hyper-iterations, the total runtime is less than full RMD with 5000 iterations, and the results are significantly improved. We also find that while using a short horizon ($T = 1$) is faster, it achieves a lower accuracy at the same number of iterations.

Finally, we verify some of our theorems in practice. Figure 2.7 (fourth plot) shows that when the lower-level problem is regularized, the relative ℓ_2 error between the K-RMD approximate gradient and the exact gradient decays exponentially as K increases. This

Table 2.4: Results for one-shot learning on Omniglot dataset. K-RMD reaches similar performance as full RMD, is considerably faster, and requires less memory.

Method	**Accuracy**	**iter.**	**Sec/iter.**
1-RMD	95.6	5K	0.4
10-RMD	96.3	5K	0.7
25-RMD	96.1	5K	1.3
Full RMD	95.8	5K	2.2
1-RMD	97.7	15K	0.4
10-RMD	97.8	15K	0.7
Short horizon	96.6	15K	0.1

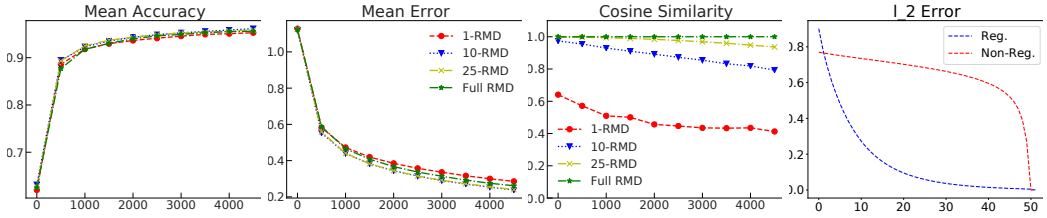


Figure 2.7: Omniglot results. **Plots 1 and 2:** Test accuracy and val. error vs. number of hyper-iterations for different RMD depths. K-RMD methods show similar performance as the full RMD. **Plot 3:** Cosine similarity between inexact gradient and full RMD over hyper-iterations. **Plot 4:** Relative ℓ_2 error of inexact gradient and full RMD vs. reverse depth. Regularized version shows exponential decay.

was guaranteed by Proposition 2.3.1. However, this exponential decay is not seen for the non-regularized model ($\rho = 0$). This suggests that the local strong convexity assumption is essential in order to have exponential decay in practice. Figure 2.7 (third plot) shows the cosine similarity between the inexact gradient and full gradient over the course of meta-training. Note that the cosine similarity measures are always positive, indicating that the inexact gradients are indeed descent directions. It also seems that the cosine similarities show a slight decay over time.

2.5 Conclusion

We analyze K-RMD, a first-order heuristic for solving bilevel optimization problems when the lower-level optimization is itself approximated in an iterative way. We show that K-

RMD is a valid alternative to full RMD from both theoretical and empirical standpoints. Theoretically, we identify sufficient conditions for which the hyperparameters converge to an approximate or exact stationary point of the upper-level objective. The key observation is that when $\hat{w}^*$ is near a strict local minimum of the lower-level objective, gradient approximation error decays exponentially with reverse depth. Empirically, we explore the properties of this optimization method with four proof-of-concept experiments. We find that although exact convergence appears to be uncommon in practice, the performance of K-RMD is close to full RMD in terms of application-specific metrics (such as generalization error). It is also roughly twice as fast. These results suggest that in hyperparameter optimization or meta learning applications with memory constraints, truncated back-propagation is a reasonable choice.

Our experiments use a modest number of parameters M, hyperparameters N, and horizon length T. This is because we need to be able to calculate both K-RMD and full RMD in order to compare their performance. One promising direction for future research is to use K-RMD for bilevel optimization problems that require powerful function approximators at both levels of optimization. Truncated RMD makes this approach feasible and enables comparing bilevel optimization to other meta-learning methods on difficult benchmarks.

CHAPTER 3

ONE-SHOT LEARNING FOR SEMANTIC SEGMENTATION

Deep Neural Networks are powerful at solving classification problems in computer vision. However, learning classifiers with these models requires a large amount of labeled training data, and recent approaches have struggled to adapt to new classes in a data-efficient manner. There is interest in quickly learning new concepts from limited data using one-shot learning methods Kaiser et al., 2017; Santoro et al., 2016. One-shot image classification is the problem of classifying images given only a single training example for each category Koch, 2015; Vinyals et al., 2016.

We propose to undertake *One-Shot Semantic Image Segmentation*. Our goal is to predict a pixel-level segmentation mask for a semantic class (like horse, bus, *etc*.) given only a single image and its corresponding pixel-level annotation. We refer to the image-label pair for the new class as the support set here, but more generally for k-shot learning, support set refers to the k images and labels.

A simple approach to performing one-shot semantic image segmentation is to fine-tune a pre-trained segmentation network on the labeled image Caelles et al., 2017b. This approach is prone to over-fitting due to the millions of parameters being updated. It also introduces complications in optimization, where parameters like step size, momentum, number of iterations, *etc*. may be difficult to determine. Recent one-shot image categorization methods Koch, 2015; Vinyals et al., 2016 in contrast, meta-learn a classifier that, when conditioned on a few training examples, can perform well on new classes. Since Fully Convolutional Neural Networks (FCNs) Long, Shelhamer, and Darrell, 2015 perform segmentation as pixel-wise classification, we could extend these one-shot methods directly to classify at the pixel level. However, thousands of dense features are computed from a single image and one-shot methods do not scale well to this many features. We illustrate this

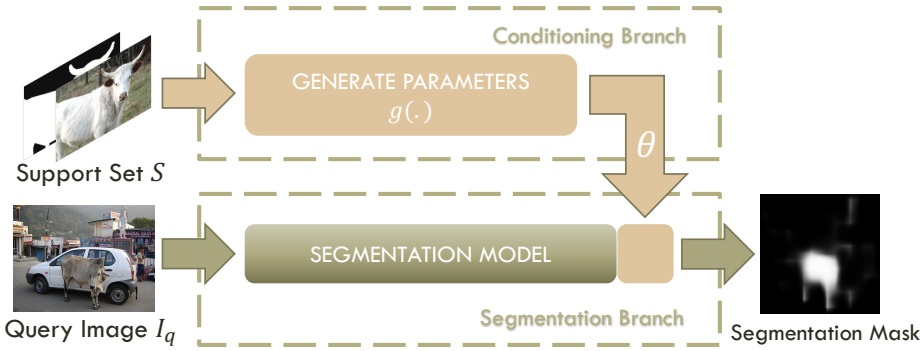


Figure 3.1: Overview. S is an annotated image from a new semantic class. In our approach, we input S to a function g that outputs a set of parameters θ. We use θ to parameterize part of a learned segmentation model which produces a segmentation mask given I_q.

issue by implementing an extension to the Siamese Network from Koch, 2015 as a baseline in Section 3.5.

We take inspiration from few-shot learning and propose a novel two-branched approach to one-shot semantic image segmentation. The first branch takes the labeled image as input and produces a vector of parameters as output. The second branch takes these parameters as well as a new image as input and produces a segmentation mask of the image for the new class as output. This is illustrated in Figure 3.1. Unlike the fine tuning approach to one-shot learning, which may require many iterations of SGD to learn parameters for the segmentation network, the first branch of our network computes parameters in a single forward pass. This has several advantages: the single forward pass makes our method fast; our approach for one-shot learning is fully differentiable, allowing the branch to be jointly trained with the segmentation branch of our network; finally, the number of parameters θ is independent of the size of the image, so our method does not have problems in scaling.

To measure the performance for one-shot semantic segmentation we define a new benchmark on the PASCAL VOC 2012 dataset *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results* (Section 3.4). The training set contains labeled images from a subset of the PASCAL classes and the testing set has annotations of classes that were not present in training. We show significant improvements over the baselines on this benchmark

in terms of the standard meanIoU (mean Intersection over Union) metric as described in Section 3.6.

We extend to k-shot learning by applying our one-shot approach for each of the k images independently to produce k segmentation masks. We then aggregate these masks by performing a logical-OR operation at the pixel level. This approach, apart from being easy to implement and fast, requires no retraining to generalize to any number of images in the support set. We show its effectiveness in terms of increasing meanIOU accuracy per added image to the support set in section 3.6.

PASCAL VOC contains only 20 classes, which is small when compared to standard datasets used for training one-shot classification methods like Omniglot (1623) Lake, Salakhutdinov, and Tenenbaum, 2015 and ImageNet (1000) (Deng et al., 2009a). Simulating the one-shot task during training, even with such a limited number of classes performs well. This is in contrast to the common notion that training models for few-shot learning requires a large number of classes. We hypothesize that part of our algorithm's ability to generalize well to unseen classes comes from the pre-training performed on ImageNet, which contains weak image-level annotations for a large number of classes. We perform experiments on the pretraining in section 3.6.1.

We make the following contributions: (1) we propose a novel technique for one-shot segmentation which outperforms baselines while remaining significantly faster; (2) we show that our technique can do this without weak labels for the new classes; (3) we show that meta-learning can be effectively performed even with only a few classes having strong annotations available; and (4) we set up a benchmark for the challenging k-shot semantic segmentation task on PASCAL.

3.1 Related Work

Semantic Image Segmentation is the task of classifying every pixel in an image into a predefined set of categories. Convolutional Neural Network (CNN) based methods have

driven recent success in the field. Some of these classify super-pixels Girshick et al., 2014; Hariharan et al., 2014; Mostajabi, Yadollahpour, and Shakhnarovich, 2015, others classify pixels directly Chen et al., 2016a; Hariharan et al., 2015; Long, Shelhamer, and Darrell, 2015; Noh, Hong, and Han, 2015. We base our approach on the Fully Convolutional Network (FCN) for Semantic Segmentation Long, Shelhamer, and Darrell, 2015 which showed the efficiency of pixel-wise classification. However, unlike FCN and the other approaches above, we do not assume a large set of annotated training data for the test classes.

Weak Supervision. Weak and semi-supervised methods for Semantic Segmentation reduce the requirement on expensive pixel-level annotations, thus attracting recent interest. Weak supervision refers to training from coarse annotations like bounding boxes Dai, He, and Sun, 2015 or image labels Papandreou et al., 2015; Pathak et al., 2014; Pinheiro and Collobert, 2015. A notable example is co-segmentation, where the goal is to find and segment co-occurring objects in images from the same semantic class Faktor and Irani, 2013; Rother et al., 2006. Many co-segmentation algorithms Chen, Shrivastava, and Gupta, 2014; Hochbaum and Singh, 2009; Quan et al., 2016 assume object visual appearances in a batch are similar and either rely on hand-tuned low-level features or high-level CNN features trained for different tasks or objects Quan et al., 2016. In contrast, we meta-learn a network to produce a high-level representation of a new semantic class given a single labeled example. Semi-supervised approaches Hong, Noh, and Han, 2015; Hong et al., 2016; Papandreou et al., 2015 combine weak labels with a small set of pixel-level annotations. However, they assume a large set of weak labels for each of the desired objects. For instance, Pathak *et al*. Pathak, Krahenbuhl, and Darrell, 2015 use image-level annotations for all classes and images in the PASCAL 2012 training set *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*, while we exclude all annotations of the testing classes from the PASCAL training set.

Few-Shot Learning algorithms seek to generalize knowledge acquired through classes seen

during training to new classes with only a few training examples Li, Fergus, and Perona, 2006; Salakhutdinov, Tenenbaum, and Torralba, 2012; Vinyals et al., 2016. Discriminative methods in which the parameters of the base classifier (learned on training classes) are adapted to the new class Bart and Ullman, 2005; Bertinetto et al., 2016; Hariharan and Girshick, 2016; Wang et al., 2016 are closely related to our work. The main challenge is that the adapted classifier is prone to over-fit to the newly presented training examples. Wang and Herbert Wang et al., 2016 address this challenge by learning to predict classifiers which remain close to the base classifier. Bertinetto *et al*. Bertinetto et al., 2016 trained a two-branch network, in which one branch receives an example and predicts a set of dynamic parameters. The second branch classifies the query image using the dynamic parameters along with a set of learned static parameters. A similar approach was used by Noh *et al*. in Noh, Hongsuck Seo, and Han, 2016 for question answering. We draw several ideas from these papers and adapt them for the task of dense classification to design our model. Metric learning is another approach to low-shot learning Koch, 2015; Vinyals et al., 2016. It aims to learn an embedding space that pulls objects from the same categories close, while pushing those from different categories apart. Koch *et al*. Koch, 2015 show that a Siamese architecture trained for a binary verification task can beat several classification baselines in k-shot image classification. We adapt their approach for image segmentation as one of our baselines.

3.2 Problem Setup

Let the support set $S = \{(I_s^i, Y_s^i(l))\}_{i=1}^k$ be a small set of k image-binary mask pairs where $Y_s^i \in L_{test}^{H \times W}$ is the segmentation annotation for image I_s^i and $Y_s^i(l)$ is the mask of the i^{th} image for the semantic class $l \in L_{test}$. The goal is to learn a model $f(I_q, S)$ that, when given a support set S and query image I_q, predicts a binary mask $\hat{M}_q$ for the semantic class l. An illustration of the problem for $k = 1$ is given Figure 3.1.

During training, the algorithm has access to a large set of image-mask pairs $D =$

$\{(I^j, Y^j)\}_{j=1}^{N}$ where $Y^j \in L_{train}^{H \times W}$ is the semantic segmentation mask for training image I^j. At testing, the query images are only annotated for new semantic classes i.e. $L_{train} \cap L_{test} = \varnothing$. This is the key difference from typical image segmentation where training and testing classes are the same. While the problem is similar to k-shot learning, which has been extensively studied for image classification Salakhutdinov, Tenenbaum, and Torralba, 2012; Vinyals et al., 2016, applying it to segmentation requires some modification.

In this problem, unlike image classification, examples from L_{test} might appear in training images. This is handled naturally when an annotator unaware of some object class, labels it as background. Annotations of L_{test} objects are excluded from the training set, while the images are included as long as there is an object from L_{train} present. State-of-the-art algorithms for image segmentation Chen et al., 2014, 2016b use networks pre-trained on large-scale image classification datasets like Deng et al., 2009a. Although these Weights give the models a better starting point, they still require many segmented images and thousands of weight updates to learn a good model for pixel classification. This is true even for the classes that directly overlap. We allow similar access to weak annotations for our problem by initializing VGG with weights pre-trained on ImageNet Deng et al., 2009a. In section 3.6.1 however, we show that even excluding all the overlapping classes from pre-training does not degrade the performance of our approach.

3.3 Proposed Method

We propose an approach where the first branch receives as input a labeled image from the support set S and the second branch receives the query image I_q. In the first branch, we input the image-label pair $S = (I_s, Y_s(l))$ to produce a set of parameters,

$$w, b = g_\eta(S). \tag{3.1}$$

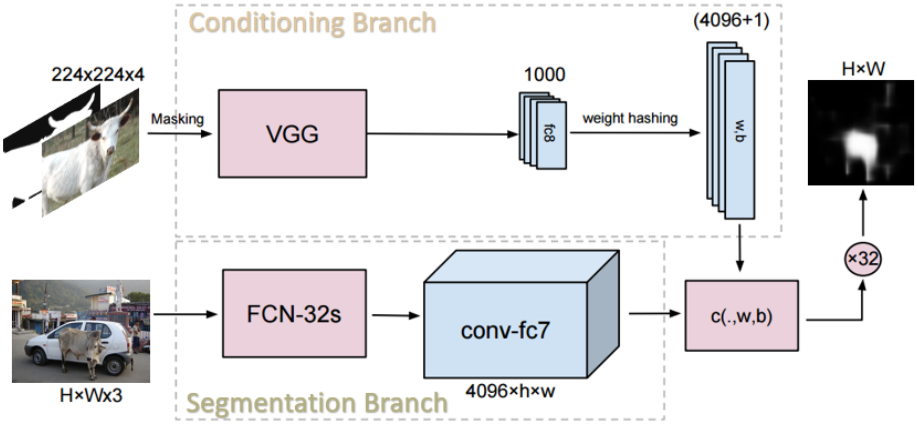


Figure 3.2: Model Architecture. The conditioning branch receives an image-label pair and produces a set of parameters $\{w, b\}$ for the logistic regression layer $c(\cdot, w, b)$. The segmentation branch is an FCN that receives a query image as input and outputs strided features of conv-fc7. The predicted mask is generated by classifying the pixel-level features through $c(\cdot, w, b)$, which is then upsampled to the original size.

In the other branch, we extract a dense feature volume from I_q using a parametric embedding function ϕ. Let $F_q = \phi_\zeta(I_q)$ be that feature volume extracted from I_q, then F_q^{mn} is the feature vector at the spatial location (m, n). Pixel level logistic regression is then performed on the features using the parameters from the first layer to get the final mask,

$$\hat{M}_q^{mn} = \sigma(w^\top F_q^{mn} + b). \tag{3.2}$$

Here, $\sigma(.)$ is the sigmoid function and $\hat{M}_q^{mn}$ is the (m, n) location of the predicted mask for the query. This can be understood as a convolutional layer with parameters $\{w, b\}$ followed by a sigmoid activation function, where the parameters are not fixed after training and get computed through the first branch for each image in the support set. The predicted mask is then upsampled back to the original image size using standard bilinear interpolation. The final binary mask is produced by using a threshold of 0.5 on $\hat{M}_q$. The overall architecture is illustrated in Figure 3.2. We explain each part of the architecture in more detail in the following subsections.

3.3.1 Producing Parameters from Labeled Image

We modify the VGG-16 architecture from Simonyan and Zisserman, 2014 to model the function $g_\eta(\cdot)$.

Masking. We chose to mask the image with its corresponding label so it contains only the target object instead of modifying the first layer to receive the four channel image-mask pair as input. We do this for the following two empirical reasons. (1) Even in the presence of the mask the network response tends to be biased towards the largest object in the image which may not be the object we would like to segment. (2) Including the background information in the input increased the variance of the output parameters $\{w, b\}$ which prevented the network from converging.

Weight Hashing. Inspired by Noh *et al*. Noh, Hongsuck Seo, and Han, 2016, we employed the weight hashing layer from Chen et al., 2015 to map the 1000-dimensional vector output from the last layer of VGG to the 4097 dimensions of $\{w, b\}$. This mapping avoids the overfitting which would occur due to the massive number of extra parameters that a fully connected layer will introduce if used instead. We implemented it efficiently as a fully connected layer with fixed weights. This is explained in more detail in the supplementary material.

3.3.2 Dense Feature Extraction

We model the embedding function $F_q = \phi_\zeta(I_q)$ by the FCN-32s fully convolutional architecture Long, Shelhamer, and Darrell, 2015 excluding the final prediction layer. The 4096 channel feature volume at conv-fc7 is then fed to the logistic pixel classifier described above. In section 3.6 we also evaluate performance of the high resolution dilated-FCN Yu and Koltun, 2015 with stride 8.

3.3.3 Training Procedure

We simulate the one shot task during training by sampling a support set S, a query image I_q and its corresponding binary mask M_q from the training set D_{train} at each iteration. First, an image-label pair (I_q, Y_q) is sampled uniformly at random from D_{train}, then we sample a class $l \in L_{train}$ uniformly from the classes present in the semantic mask and use it to produce the binary mask $Y_q(l)$. S is formed by picking one image-mask pair at random from $D_{train} - \{(I_q, Y_q)\}$ with class l present. We can then predict the mask $\hat{M}_q$ with a forward pass through our network. We maximize the log likelihood of the ground-truth mask

$$\mathcal{L}(\eta, \zeta) = \mathop{\mathbb{E}}_{S, I_q, M_q \sim D_{train}} \Big[\sum_{m,n} \log p_{\eta,\zeta}(M_q^{mn} | I_q, S) \Big]. \tag{3.3}$$

Here η and ζ are the network parameters, $p_{\eta,\zeta}$ is the probability of the mask given the neural network output, and S, I_q, and M_q are sampled by the sampling strategy described above. We use Stochastic Gradient Descent with a fixed learning rate of 10^{-10}, momentum 0.99 and batch size of 1. The VGG network overfits faster than the fully-convolutional branch; therefore, we set the learning rate multiplier to 0.1 for learning the parameter η. We stop training after 60k iterations.

3.3.4 Extension to k-shot

In the case of k-shot segmentation the support set contains k labeled images, $S = \{I_s^i, Y_s^i(l)\}_{i=1}^k$. We use these images to produce k sets of the parameters $\{w^i, b^i\}_{i=1}^k$. Each of them can be understood to be an independent classifier of an ensemble. These classifiers we noticed have high precision but low recall. We believe this is because each is produced by one example from the support set and a single image can only contain a small subset of the possible appearances of the object. So, we combine the decision of these classifiers by including a pixel in the final mask if it was considered an object by any of these classifiers. This is implemented as a logical OR operation between the k binary masks. This approach has

the benefit that it does not require any retraining and can be generalized to any k. It is also much faster than the baselines as shown in section 3.6.

3.4 Dataset and Metric

Dataset: We create a new dataset, **PASCAL-$5^{\mathbf{i}}$**, for the problem of k-shot Image Segmentation using images and annotations from PASCALVOC 2012 *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results* and extended annotations from SDS[1] Hariharan et al., 2014. From L, the set of twenty semantic classes in PASCALVOC, we sample five and consider them as the test label-set $L_{test} = \{4i+1, \ldots, 4i+5\}$, with i being the fold number, and the remaining fifteen forming the training label-set L_{train}. Test and training class names are shown in Table 3.1. We form the training set D_{train} by including all image-mask pairs from PASCALVOC and SDS training sets that contain at least one pixel in the semantic mask from the label-set L_{train}. The masks in D_{train} are modified so that any pixel with a semantic class $\neq L_{train}$ is set as the background class $l_{\varnothing}$. We follow a similar procedure to form the test set D_{test}, but here the image-label pairs are taken from PASCALVOC validation set and the corresponding label-set is L_{test}. Thus, apart from a few exclusions, the set of images is similar to those used in Image Segmentation papers, like FCN Long, Shelhamer, and Darrell, 2015. However, the annotations are different. Given the test set D_{test}, we use the same procedure that is described in Section 3.3.3 to sample each test example $\{S, (I_q, Y_q(l))\}$. We sample $N = 1000$ examples and use it as the benchmark for testing each of the models described in the next section.

Metric: Given a set of predicted binary segmentation masks $\{\hat{M}_q\}_{i=1}^{N}$ and the ground truth annotated mask $\{M_q\}_{i=1}^{N}$ for a semantic class l we define the per-class Intersection over Union (IoU_l) as $\frac{tp_l}{tp_l+fp_l+fn_l}$. Here, tp_l is the number of true positives, fp_l is the number of false positives and fn_l is the number of false negatives over the set of masks. The **meanIoU** is just its average over the set of classes, i.e. $(1/n_l)\sum_l IoU_l$. This is the standard metric

[1]For creating the training set, we only include images that do not overlap with the PASCALVOC 2012 validation set.

Table 3.1: **PASCAL-5^i** test classes. For the i^{th} fold, we exclude the corresponding testing classes and pick examples from PACAL training set on the remaining $15 + 1$ classes. For testing, we pick examples from test classes in the PASCAL validation. Thus, both classes and the data are different in training and testing.

$i = 0$	$i = 1$	$i = 2$	$i = 3$
aeroplane, bicycle, bird, boat, bottle	bus, car, cat, chair, cow	diningtable, dog, horse, motorbike, person	potted plant, sheep, sofa, train, tv/monitor

Table 3.2: Mean IoU results on PASCAL-5^i. The **top** and **bottom** tables contain the semantic segmentation meanIoU on all folds for the 1-shot and 5-shot tasks respectively.

Methods (1-shot)	PASCAL-5^0	PASCAL-5^1	PASCAL-5^2	PASCAL-5^3	Mean
1-NN	25.3	44.9	**41.7**	18.4	32.6
LogReg	26.9	42.9	37.1	18.4	31.4
Finetuning	24.9	38.8	36.5	30.1	32.6
Siamese	28.1	39.9	31.8	25.8	31.4
Ours	**33.6**	**55.3**	40.9	**33.5**	**40.8**

Methods (5-shot)	PASCAL-5^0	PASCAL-5^1	PASCAL-5^2	PASCAL-5^3	Mean
Co-segmentation	25.1	28.9	27.7	26.3	27.1
1-NN	34.5	53.0	**46.9**	25.6	40.0
LogReg	**35.9**	51.6	44.5	25.6	39.3
Ours	**35.9**	**58.1**	42.7	**39.1**	**43.9**

of meanIU defined in Image Segmentation literature adapted for our binary classification problem.

3.5 Baselines

We evaluate the performance of our method with different baselines. Since one-shot image segmentation is a new problem, we adapt previous work for dense pixel prediction to serve as baselines to compare against.

- **Base Classifiers:** CNNs learn deep representations of images, so these models are an intuitive starting point for classification. Specifically, we first fine-tune FCN-32s pretrained on ILSVRC2014 data to perform 16-way (15 training foreground classes + 1 background class) pixel-wise predictions on the **PASCAL-5^i** dataset. During testing, we extract dense pixel-level features from both images in the support set and

the query image. We then train classifiers to map dense fc-7 features from the support set to their corresponding labels and use it to generate the predicted mask $\hat{M}_q$. We experimented with various classifiers including 1-NN and logistic regression[2]

- **Fine-tuning:** As suggested by Caelles et al., 2017b, for each test iteration we fine-tune the trained segmentation network on examples in the support set and test on the query image. We only fine-tune the fully connected layers (fc6, fc7, fc8) to avoid overfitting and reducing the inference time per query. We also found that the fine-tuned network converges faster if we normalize the fc-7 features by a batch normalization layer.

- **Co-segmentation by Composition:** To compare with the these techniques, we include the results of the publicly available implementation[3] of Faktor and Irani, 2013 on PASCAL-5^i.

- **Siamese Network for One-shot Dense Matching:** Siamese Networks trained for image verification, i.e. predicting whether two inputs belong to the same class, have shown good performance on one-shot image classification Koch, 2015. We adapt them by using two FCNs to extract dense features and then train it for pixel verification. A similarity metric from each pixel in the query image to every pixel in the support set is also learned and pixels are then labeled according to their nearest neighbors in the support set. Implementation details are provided in the supplementary document.

3.6 Experiments

We conduct several experiments to evaluate the performance our approach on the task of k-shot Image segmentation by comparing it to other methods. Table 3.2 reports the performance of our method in 1-shot and 5-shot settings and compares them with the baseline methods. To fit a 5-shot Siamese network into memory we sampled from features

[2]We also trained linear SVM, but could not get a comparable results to logistic regression.
[3]http://www.wisdom.weizmann.ac.il/ vision/CoSegmentationByComposition.html.

in the support set with a rate of 0.3. However, sub-sampling considerably degraded the performance of the method and 5-shot results were worse than the 1-shot version so we exclude those results.

Our method shows better generalization performance to new classes. The difference is very noticeable in 1-shot learning as other methods overfit to only the image in the support set. Specifically, our method outperforms 1-NN and fine-tuning in one-shot image segmentation by 25% relative meanIoU. We also provide some qualitative result from our method in Figure 3.4. Surprisingly, the results for 1-NN are almost as good as the fine-tuning baseline, which overfits quickly to the data in the support set.

In Table 3.2, we also compare Co-segmentation by Composition Rother et al., 2006 for 5-shot segmentation to our approach. As expected, using the strong pixel-level annotations enables our method to outperform the unsupervised co-segmentation approach, by 16%. In fact, we can outperform co-segmentation results that require 5 weakly annotated images with just a single strongly annotated image.

Dilated-FCN: In addition to the low-resolution version of our method, we also trained the dilated-FCN with higher resolution on **PASCAL- 5^0** and achieved 37.0% and 37.43% meanIoU for 1-shot and 5-shot respectively. We notice a 3.4% improvement over low-resolution for one-shot, however, the gap between 1-shot and 5-shot is small at this resolution. We believe this is due to our training being specific to the 1-shot problem. We do not use dilated-FCN architecture for other methods due to the impracticality caused by their high computational cost or memory footprint.

Running Time: In Table 3.3 we include the running time of each algorithm. All the experiments were executed on a machine with a 4GHz Intel Core-i7 CPU, 32GB RAM, and a Titan X GPU. In one-shot setting our method is $\sim 3\times$ faster the than second fastest method logistic regression. For 5-shot our method is $\sim 10\times$ faster than logistic regression.

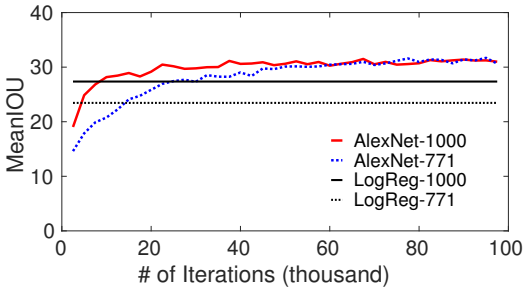


Figure 3.3: Pretraining Effect on AlexNet.

Table 3.3: Inference Time (in s).

Methods	1-shot	5-shot
1-NN	1.10	4.55
Logistic Reg	0.66	3.50
Finetune	5.56	-
Siamese	5.65	-
Ours-32s	0.19	0.21

3.6.1 Pretraining Effect

The models compared above have two sources of information, the image-level labels for the classes in ImageNet Deng et al., 2009a through the pretraining and the pixel-level annotation of classes in L_{train}. Although the test classes L_{test} do not overlap with L_{train}, they have partial overlap with some ImageNet classes. To understand this effect, we use a dataset which excludes all the classes in ImageNet with any overlap with PASCAL categories called PASCAL-removed-ImageNet as in Huh, Agrawal, and Efros, 2016. This dataset contains only 771 classes as compared to 1000 originally since each class in PASCAL usually overlaps with multiple ImageNet classes. We use AlexNet Krizhevsky, Sutskever, and Hinton, 2012 trained on ImageNet and PASCAL-removed-ImageNet (from Huh *et al*. Huh, Agrawal, and Efros, 2016) with the suffices 1000 and 771 respectively. We replaced the VGG and FCN from both branches of our approach with AlexNet to give us AlexNet-1000 and AlexNet-771. We also have a baseline in the form of Logistic Regression performed on convolutional AlexNet features finetuned on PASCAL, similar to the Base Classifiers described in section 3.5. We refer to these as LogReg-1000 and LogReg-771. Figure 3.3

contains the results for these models on the first fold, i.e. PASCAL-5^0. Note that the results for the two baselines are constant because we evaluate the networks only once they converge.

In Figure 3.3 we observe that AlexNet-1000 is better initially and shows faster convergence. However, after convergence AlexNet-771 performs on par with AlexNet-1000. The initial gap could be understood by the fact that even the L_{train} classes were not presented during the pre-training. AlexNet being a simpler model performs worse than VGG, meanIOU was 33.6% in Table 3.2. However, AlexNet-771 outperforms even our best VGG baseline, which was Siamese at 28.1% for PASCAL-5^0. This result shows that we can generalize to new categories without any weak supervision for them. In contrast, LogReg-1000 outperforming LogReg-771 shows its incapacity to learn a good representation without seeing weak labels for test categories. This highlights the importance of meta-learning for this task.

3.7 Conclusion

Deep learning approaches have achieved top performance in many computer vision problems. However, learning a new concept given few examples is still a very challenging task. In this chapter we developed a new architecture to address this problem for image segmentation. Our architecture learns to learn an ensemble classier and use it to classify pixels in the query image. Through comprehensive experiments we show the clear superiority of our algorithm. The proposed method is considerably faster than the other baselines and has a smaller memory footprint.

Figure 3.4: Some qualitative results of our method for 1-shot. Inside each tile, we have the support set at the top and the query image at the bottom. The support is overlaid with the ground truth in yellow and the query is overlaid with our predicted mask in red.

CHAPTER 4

VIDEO SEGMENTATION WITH ONE-SHOT OBJECT PROPOSALS

Video object segmentation, the task of consistently separating foreground object(s) from the background in video, is a fundamental problem in computer vision. Video segmentation is a key component in applications such as action recognition, video summarization, and movie post-production. The diverse set of applications has led to different problem definitions and algorithms. Most of these algorithms can be categorized into three types based on the level of required annotation: 1) unsupervised algorithms (Li et al., 2013; Wu et al., 2015; Xiao and Jae Lee, 2016) automatically extract a pool of volumetric object proposals without annotation. User input might be required later to select useful proposals from the pool; 2) semi-supervised algorithms (Caelles et al., 2017a; Perazzi et al., 2017; Voigtlaender and Leibe, 2017) assume objects are annotated in the first frame; 3) interactive segmentation algorithms provide an interactive interface that repeatedly receives user input to enhance the segmentation results. While user input is crucial to reduce the ambiguity in the video segmentation task, a segmentation algorithm should ideally minimize user effort. Researchers have focused on different sources of information that can minimize human effort in the task. Instance-aware semantic segmentation algorithms (He et al., 2017; Li et al., 2016b) can provide annotations for most of the objects in a scene. However, performance degrades for unseen object classes. Recently, one-shot segmentation algorithms (Caelles et al., 2017a; Khoreva et al., 2017; Shaban et al., 2017a; Voigtlaender and Leibe, 2017), which learn a model from object annotations in the first frame, show state-of-the-art performance in semi-supervised video object segmentation tasks. However, these methods require annotation of all the target objects in the first frame.

An ideal scenario would be that the algorithm segments most objects automatically, give the user the ability to select/reject from the sparse set of proposed tracks. If the algorithm

Figure 4.1: Our approach can segment and track more than 15 objects in this video, including many people in the audience, while the dataset only has 3 of them annotated

fails to segment some objects, user could optionally provide semi-supervised annotation for the missing ones (Fig.4.1). This process would be efficient since the user may only need to check the first frame to accept/reject tracks or annotate objects, and does not have to annotate everything in the first frame. In this chapter, we achieve such flexibility by having separate stages for image proposal extraction and unsupervised proposal tracking, which can utilize proposals generated from heterogeneous sources. We show that an instance-aware semantic segmentation algorithm, FCIS (Li et al., 2016b), can cover most of the objects in the videos. For the objects with annotations provided by the user, we enhance one-shot object segmentation approach in (Caelles et al., 2017a) to obtain sequence-specific object proposals. Through the experiments, we show that the maximum performance of the algorithm is reached when few objects that do not come from a semantic category are annotated and the rest are covered by the semantic proposals.

Our segment tracking approach is a novel, enhanced version of the multi-segment tracking and object discovery algorithm SPT (Li et al., 2013; Wu et al., 2015), which learns a long-term holistic appearance model on segment proposals based on least-squares optimization. SPT finds and tracks objects that start from any frame and last for any duration, learns a long-term appearance model for each object, handles partial and complete occlusions and finds completely occluded objects that re-enter the scene. However, its tracks

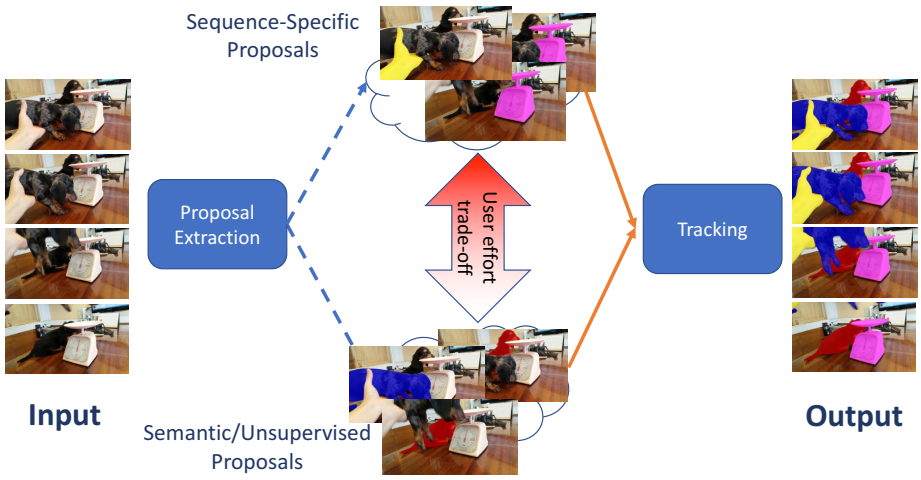


Figure 4.2: **Overall Architecture** The proposed algorithm extracts two types of proposals from the input frames. Semantic proposals provide annotations for most of the objects in the scene without significant user effort. Users can select/reject proposed tracks by checking the first frame, or provide the annotation for the desired instances to get improved performance for missed objects. We use user annotations to extract sequence-specific proposals. The tracking algorithm receives all the proposals and learns long-term temporal models to track segments. The tracker focuses on handling occlusion, motion model, and forward/backward tracking to improve the performance.

can be a bit noisy; in our enhanced SPT-Retrack model, all the found object tracks after SPT are backtracked to the 1st frame. After consolidating the tracks with a motion model, SPT is used again to learn a long-term appearance model of the consolidated tracks, which improves performance. Finally, we utilize the spatial refinement network in Shaban et al., 2017a that refines the the pixel level object confidence map output of SPT.

Figure 4.2 shows the overall architecture of the proposed approach. The main contribution of this work is the combination of powerful ideas from instance segmentation and unsupervised/semi-supervised object video segmentation into a unified framework. There are a number of novelties in our approach: 1) An extension of OSVOS (Caelles et al., 2017a) generates semi-supervised sequence-specific segment proposals; 2) we propose SPT-Retrack which uses backtracking and re-tracking to enhance segment tracks found by SPT; and 3) we use a novel spatial refinement network based on deep image matting to enhance final predictions.

4.1 Related Work

Video segmentation has received a lot of interest in recent years. A popular semi-supervised version of video segmentation assumes that the ground truth is available in the first frame (Märki et al., 2016). DAVIS-2016 (Perazzi et al., 2016) and the recently published DAVIS-2017 dataset (Pont-Tuset et al., 2017) are state-of-the-art benchmarks for this problem. While DAVIS-2016 has a single foreground object in each video, in DAVIS-2017 more challenging sequences with multiple objects are presented. Semi-supervised video segmentation is also related to unsupervised foreground/background segmentation algorithms (Faktor and Irani, 2014; Papazoglou and Ferrari, 2013) in a dataset with only one foreground moving object, which is the case for DAVIS-2016, but not some other benchmarks such as SegTrack v2 (Li et al., 2013), which has both moving and still (or slow-moving) objects annotated.

Deep neural networks have recently demonstrated state-of-the-art performance in semi-supervised video segmentation. Mask propagation networks (Perazzi et al., 2017) are trained to predict the object mask given the current frame and the previous prediction. In one-shot video object segmentation networks (Caelles et al., 2017a) (OSVOS) the first annotated frame is treated as a training example and the model is fit to that. Recently, the lucid-dream algorithm (Khoreva et al., 2017) showed state-of-the-art performance in the DAVIS-2017 challenge (Pont-Tuset et al., 2017) by fine-tuning the mask propagation network on augmented images from the first frame. The application of both of these methods is limited to the cases where strong annotation for all the objects are available in the first frame. We borrow ideas from these algorithms to generate proposals for the subset of objects which are annotated in the first frame.

There is also recent work on moving object proposals (Fragkiadaki et al., 2015), semantic segmentation (Long, Shelhamer, and Darrell, 2015) and semantic video segmentation tasks (Shelhamer et al., 2016). Recently, many efforts have been made to incorporate time into the models. 3-dimensional convolutional models are used to learn spatiotemporal

features for video classification and activity recognition (Ji et al., 2013; Karpathy et al., 2014). Recurrent architecture such as LSTM (Hochreiter and Schmidhuber, 1997) maintains and updates the state of the network in each time step and can model long-term dependencies in the data. More recently, Shelhamer *et al*. (Shelhamer et al., 2016) propose clockwork FCN which has different update rates for each FCN layer, and use it for the video semantic segmentation task.

A closely related research field is instance-aware semantic segmentation (He et al., 2017; Li et al., 2016b). State-of-the-art methods utilize output from region proposal networks (Ren et al., 2015) (RPNs) to generate instance-level segmentation masks for the objects. In fully convolutional instance-aware semantic segmentation (Li et al., 2016b) (FCIS), location-sensitive outputs are aggregated from RPN proposals and scored to generate instance masks.

4.2 Object Proposal Generation

For objects with annotations in the first frame, we generates sequence-specific proposals with a modified version of the one-shot object video segmentation model (Caelles et al., 2017a) to generate proposals for the target object. Proposals generated with this method are combined with the proposal from the state-of-the-art instance-aware image segmentation algorithm FCIS (Li et al., 2016b) which covers COCO (Lin et al., 2014a) classes.

4.2.1 Sequence-Specific Proposals

The Network Architecture: Following (Caelles et al., 2017a), we adapt the VGG16 (Simonyan and Zisserman, 2014) architecture for dense pixel prediction. Fully-connected layers are removed, and convolutional skip connections (Long, Shelhamer, and Darrell, 2015) with kernel size 3 are added before the last 4 pooling layers. The output of the skip connections are up-sampled to the size of the original image, concatenated with each other, and fed to the prediction layer. To allow the network to use motion information, we modify the input layer to receive optical flow magnitude in addition to the RGB image, as

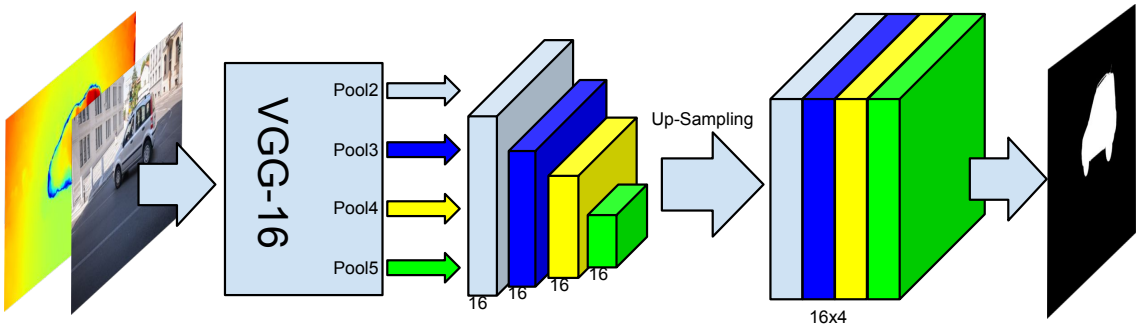


Figure 4.3: **Sequence-Specific Proposal Generator Architecture** Our network receives a 4 channel input (RGB+flow magnitute). Skip connections are taken from the just before the pooling layers illustrated in the figure. The final convolutional layer maps a 64 channel input to the prediction mask.

in (Khoreva et al., 2017). Figure 4.3 shows the overall architecture of the model. Weights are initialized with a VGG-16 model trained on the ImageNet dataset (Simonyan and Zisserman, 2014). We used the Xavier initialization (Glorot and Bengio, 2010) to initialize the new layers.

Offline Training: The network is trained on the DAVIS-2017 training dataset using stochastic gradient descent with initial learning rate 10^{-8}, batch size 10, and weight decay 10^{-5}, for 50k iterations. We used MRFlow (Wulff, Sevilla-Lara, and Black, 2017) to extract optical flow from the videos. For a specific sequence, let $\mathcal{M}_{it}$ and $\mathcal{G}_{it}$ be the prediction and the binary ground-truth mask for instance i in time t. At each iteration we pick a random instance from the dataset and execute a gradient decent update. We use a modified version of balanced cross entropy loss function in our model:

$$L_{it} = \alpha_{it} \sum_{x \in \mathcal{G}_{it}^{-}} \log \mathcal{M}_x + \beta_{it} \sum_{x \in \mathcal{G}_{it}^{+}} \log \left(1 - \mathcal{M}_x\right) \tag{4.1}$$

where $\mathcal{M}$ is the prediction, $\alpha_{it} = \frac{|\mathcal{G}_{it}^{-}|}{|\mathcal{G}_{it}|}$ and $\beta_{it} = \min(\frac{|\mathcal{G}_{it}^{+}|}{|\mathcal{M}_{it}|}, 0.1)$ where $\mathcal{G}_{it}^{-}$ and $\mathcal{G}_{it}^{+}$ are the set of background and foreground points in the mask, respectively. Setting the minimum value of 0.1 is specially crucial in the online-training step for the network to converge on small objects (see Figure 4.4).

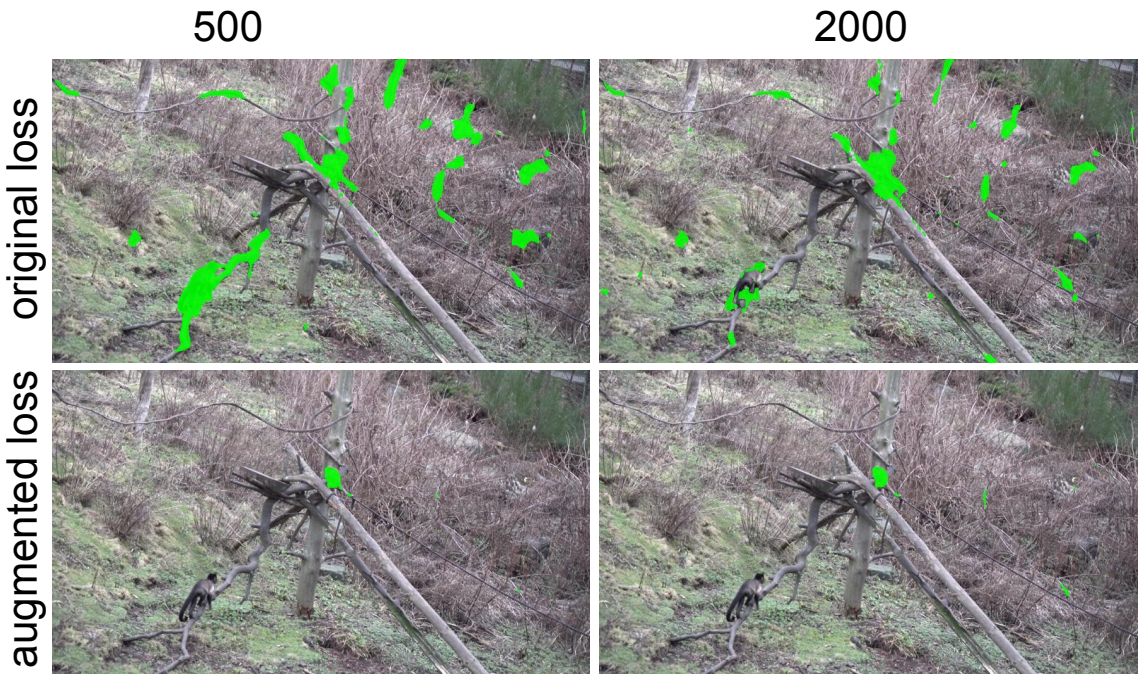


Figure 4.4: **Effect of the Loss Function** Loss function in the OSVOS algorithm (Caelles et al., 2017a) does not converge after 2000 iterations in the monkeys-trees sequence (**first row**). The augmented loss in Equation 4.1 converges after 500 iterations (**second row**)

Online training/testing To generate proposals for each instance j in the test set, we fine-tune the trained network on the first frame image/mask $(\mathcal{I}_0, \mathcal{G}_{j0})$ and test the network on the rest of the frames which gives us one proposal per frame, per object. To increase the generalization performance of the network, we augment the training set $(\mathcal{I}_0, \mathcal{G}_{j0})$ using the method described in (Khoreva et al., 2017) to generate 750 augmented images. This augmentation method also provides synthetic optical flow values to feed into the network. We fine-tune the pre-trained network for 2k iterations with the same setting as the offline-training. It is known that even with highly augmented data, the network may overfit to the appearance of the object in the first frame as well as classify similar instances as foreground (Figure 4.5 first column). In some sequences, as the target object appearances change, the predictions drifts gradually from one instance to another instance in the scene.

Combinatorial Grouping To alleviate this problem, we use a simple and effective combinatorial grouping algorithm to generate many proposals from each prediction. This

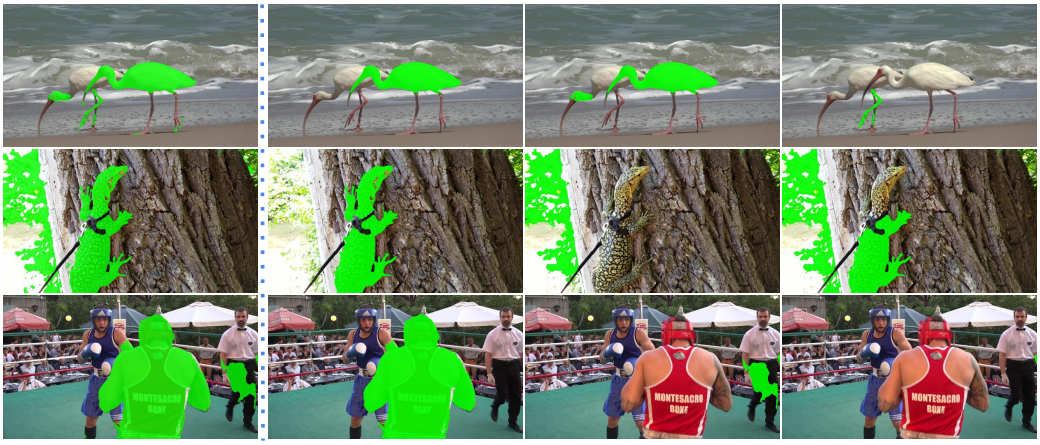


Figure 4.5: **Combinatorial Grouping** the first column shows the prediction of the network. The second column shows the best proposal generated from combinatorial grouping algorithm. Other columns show other randomly sampled proposals. The grouping algorithm increases recall by taking into account the prior on the continuity of the parts, the maximal distance between parts, and the area of each part.

method effectively removes noisy predictions with small areas that tricks our motion model and also separates similar instances that appear as foreground in the prediction. Given the binary prediction mask $\mathcal{M}_{jt}$, we first find the spatially connected components in the mask with an area of more than k pixels. Next we generate all the possible combinations of the remaining connected components. Finally, we reject any combination in which the distance between two connected component is more than d pixels. See Figure 4.5 for some qualitative illustrations. We tune the parameter of the combinatorial grouping on DAVIS-2017 validation set.

4.2.2 Sequence-Independent Proposals

Recently, instance-aware image segmentation algorithms (He et al., 2017; Li et al., 2016b; Pinheiro et al., 2016) have displayed exceptional performance in segmenting objects from known semantic classes. To cover these semantic classes, we enriched the set of proposals by the proposals generated from the FCIS (Li et al., 2016b) algorithm[1]. Since our proposals are class-agnostic, we only use the mask predictions and discard the semantic labels. FCIS

[1]We use the publicly available FCIS implementation https://github.com/msracver/FCIS.

cannot cover some categories it is not trained on, hence we also tried using category-agnostic proposal extraction methods SharpMask (Pinheiro et al., 2016) and COB (Maninis et al., 2017), but these methods resulted in inferior performance on non-COCO categories since they were also deep models trained on COCO categories. Hence, in some of the experiments we utilize POISE (Humayun, Li, and Rehg, 2015) which is the best approach for generating object proposals without any training on deep networks.

4.3 Segment Proposal Tracking

Our tracking algorithm is unique in that it does not learn from the ground truth annotation in the first frame. Instead, it is a general unsupervised video segmentation algorithm that is **tested** on the ground truth annotation in the first frame. Therefore, it does not depend on the existence of a precise annotation in the first frame, nor does it depend on the ground truth object being present in the first frame. Such flexibility can be beneficial in some applications. We extended the segment proposal tracking (SPT) algorithm(Li et al., 2013; Wu et al., 2015), with some simplifications and improvements that help to track objects more accurately. In the following, we will first briefly describe SPT, then introduce our improvements.

4.3.1 Segment Proposal Tracking

Segment proposal tracking (SPT) is based on recursive least-squares for tracking video segments. The algorithm assumes the existence of a large pool of image-level segment proposals, with some being real objects and some being random areas with no clear semantic meaning. In each frame, an appearance model is learned for each segment proposal, and these appearance models are updated in the subsequent frames, eventually creating long-term appearance models for each track that incorporate all the past appearances of each object track. A greedy matching algorithm prunes proposals that are not matching well, reducing the number of tracks to be much less than the number of proposals in each frame.

The appearance model in SPT is a regressor from an appearance feature of each segment

to the IoU between this segment and the target segment. Each track is supposed to be *represented* by a segment proposal in each frame, and hence this IoU can be computed if the representer is known. The prediction of the IoU creates a strong appearance model that can be tested in other frames. For example, if the object that is being tracked is a person, the regressor needs to be able to output 1 for a segment corresponding to the full person, and varying degrees for parts of the person (e.g. 0.3 for head, 0.2 for an arm, etc.). In other words, the model is implicitly learning all the parts and their combinations as defined by different proposals inside the person.

The recursive least squares formulation is the key to the computational efficiency of the SPT algorithm (Li et al., 2013):

$$\min_{\mathbf{W}} \|\mathbf{X}\mathbf{W} - \mathbf{V}\|_F^2 + \lambda \|\mathbf{W}\|_{\mathrm{F}}^2, \tag{4.2}$$

where the goal is to recover the $k \times o$ weight matrix $\mathbf{W}$, given $n \times k$ input matrix $\mathbf{X}$ (appearance features) and $n \times o$ spatial IoU matrix $\mathbf{V}$. n is the number of examples, k the dimensionality and o the number of distinct tracks. $\|\mathbf{W}\|_{\mathrm{F}}$ is a Frobenius norm regularization. Designate $\mathbf{H} = \mathbf{X}^\top \mathbf{X} = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top$ and $\mathbf{C} = \mathbf{X}^\top \mathbf{V} = \sum_{i=1}^{n} \mathbf{x}_i^\top \mathbf{v_i}$, where $\mathbf{x}_i$ and $\mathbf{v}_i$ are columns of $\mathbf{X}$ and $\mathbf{V}$, respectively. The solution of the least squares is given by the linear system:

$$(\mathbf{H} + \lambda \mathbf{I})\mathbf{W} = \mathbf{C}. \tag{4.3}$$

The pair $(\mathbf{H}, \mathbf{C})$ are the sufficient statistics of the least squares model. We denote $L = (\mathbf{H}, \mathbf{C})$ as a **least squares object** (LSO). Note each column in $\mathbf{C}$ corresponds to a distinct target (with the output a column in $\mathbf{V}$) but the linear system can be solved jointly.

The least squares tracker is an optimal online tracker since updates can be made on the LSO which adds examples and retains optimality. Suppose $L_{t-1} = (\mathbf{H}_{t-1}, \mathbf{C}_{t-1})$ is the LSO

from frame 1 to $t-1$, one can perform:

$$\mathbf{H}_t = \mathbf{H}_{t-1} + \mathbf{X}_t^\top \mathbf{X}_t, \quad \mathbf{C_t} = \mathbf{C}_{t-1} + \mathbf{X}_t^\top \mathbf{V}_t \tag{4.4}$$

to add examples described by $(\mathbf{X}_t, \mathbf{V}_t)$ to the regression problem. In order to remove targets, one only needs to remove the corresponding column in $\mathbf{C}_t$.

Once trained at every frame from 1 to t, all the o tracks in SPT can be tested in frame $t+1$, and each proposal in frame $t+1$ would receive predicted IoUs w.r.t. each track. A greedy matching is performed, so that the segment proposal that has the highest predicted IoU for each track is matched to the track and treated as the representative of the track. If two or more tracks match to the same proposal, only the track with the highest matching score is kept and other tracks are pruned. Such pruning effectively reduces the number of tracks by about half each frame, and it is easier for spurious proposals to be pruned since the same "bad" proposals may not consistently appear in every frame. In practice, any track that can successfully match for more than $5-7$ frames is quite likely to be a semantically meaningful object or object part.

In order to track objects that do not start from the first frame, SPT initiates a different LSO from each frame. In each frame t, object proposals that did not match any track will start their own tracks and become targets of the LSO L_t. All the LSOs are updated and matched simultaneously, if all the tracks in an LSO were pruned, then this LSO is no longer updated. In (Wu et al., 2015), a merging process was used to merge 10 different LSOs into 1, this is not adopted in this version.

The benefit of the SPT algorithm is that it is fast (only a linear system needs to be solved for tracking thousands of proposals) and is an optimal online algorithm. Since it learns long-term appearance models, it can be robust and not easily drifting. Even if occlusion occurs, SPT can often find back the object when it reappears(Wu et al., 2015). The drawback of the SPT algorithm is that it does not employ a complex motion model (only a linear

motion model on the object center was used), and that the temporal consistency is low. Temporal consistency is affected by several issues. First, the correct proposal may not be present in every frame, so the tracker may be forced to use a part or a much larger object in this frame, or drift to another one before coming back in the next frame, leading to significant flickering and occasional drifting. Second, the greedy matching algorithm may lead to a track that starts several frames after the object enters a frame (since initially the matching process is more noisy), or end a track prematurely before the object leaves the frame. Third, there is no spatial-temporal consistency term, so each frame may have different edge spurs or include/exclude certain parts of the object randomly. In highly complex datasets such as DAVIS-2017, with significant occlusions and motions, this may not achieve optimal performance.

4.3.2 SPT-Retrack

In order to improve the consistency of SPT, we propose SPT-Retrack, where tracks found by SPT are improved by backtracking and forward tracking, as well as an enhanced motion model. The assumption is that SPT is usually able to track every object in the frame given that the correct proposals are often present, and the number of tracks SPT returns is usually a lot less than the number of proposals in each frame. Therefore, once SPT has finished, the longer tracks (e.g. more than 7 frames) that are more likely to be an object can be refined by a less noisy matching process and a better motion model.

SPT-Retrack consists of few steps. These steps are described separately below:

Backtracking towards the 1-st frame At the end of the SPT algorithm, we have LSOs $L_1, \ldots, L_T$ (many may have already been pruned if there are no tracks associated with them). Each LSO L_t has a few segment tracks associated with it that started tracking from frame t. From L_T backwards, we run the SPT algorithm in reverse-time $(T, T-1, \ldots, 1)$ to backtrack to the 1-st frame. Therefore, we update L_T with the proposals at time $T-1$, then L_T and L_{T-1} with the proposals at time $T-2$, until we reach frame 1. The matching

algorithm is only used to select the representative of the track. No new tracks are created or removed during this process.

Motion-based Inference It is not possible to incorporate a strong motion model in SPT due to the algorithm simultaneously tracking thousands of objects. However, once we consolidated to fewer tracks, a stronger, pixel-level motion model can be used to adjust the score of each segment. We did not use optical flow as a motion model because empirically we found it to be too inaccurate, especially when occlusion occurs. In this work, we utilize a semi-Markov motion model with constant velocity on the segment level, defined as follows:

$$\begin{aligned} M_t(p_i) &= \frac{\sum_{j=1}^{10} w_j S_{t-j,j}(p_i)}{\sum_{p_i}\sum_{j=1}^{10} w_j S_{t-j,j}(p_i)} \\ S_{t-j,j} &= G_\sigma * T_{\mathbf{v}_{t-j}}(S_{t-j,j-1}),\ S_{t-j,0} = S_{t-j} \\ \mathbf{v}_t &= 0.6\mathbf{v}_{t-1} + 0.4(c(S_t) - c(S_{t-1})) \end{aligned} \tag{4.5}$$

where $M_t(p_i)$ is the motion factor at pixel p_i at time t, S_t is the representative segment of the track at frame t and $S_{t,j}$ is the estimated location of S_t after j frames of motion. $T_{\mathbf{v}_t}$ denotes a translation operator with $\mathbf{v}_t$ being the amount of translation (velocity), G_σ is a Gaussian blur with parameter σ. In each frame, $S_{t,j}$ is updated by applying the velocity vector $\mathbf{v}_t$ to the segment mask first, and then applying a Gaussian blur with parameter σ. The velocity vector is initialized to be the difference between the centroid of segments S_k and S_{k-1}, denoted as $c(S_t)$ and $c(S_{t-1})$, and updated at each frame with a momentum factor of 0.6. This motion model takes into account the motion in the previous 10 frames, with more distant frames blurred more and having smaller weights w_j. This accounts for the fact that tracking may be noisy and in some frames the results may be completely wrong. Finally, $M_t(p_i)$ is normalized to be a distribution.

After computing $M_t(p_i)$, the motion score of each proposal S_t in frame t is computed as

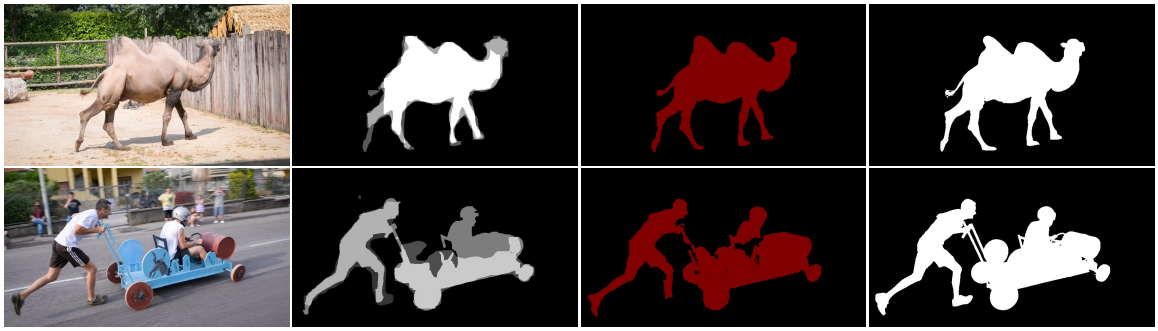


Figure 4.6: **Deep Matting Refinement.** The pixel-level weighted-averaged mask is used as the trimap into a deep matting network. The final result snaps with boundaries significantly better than the input to the matting

the average log-likelihood of all the pixels in S_t:

$$m(S_t) = \frac{\sum_{p_i \in S_t} \log(M_t(p_i))}{|S_t|} \tag{4.6}$$

where $|S_t|$ denotes the number of pixels in S_t. Then, the segment proposal that maximizes the final score $p(S_t) = o_w(S_t) + \alpha_m m(S_t)$ is selected as the segment representing the track, where $o_w(S_t)$ is the predicted overlap. Such $p(S_t)$ is stored as p_t.

Re-tracking In this step, the tracking targets are fixed to be the targets found from the first run of SPT (no greedy pruning is used). At each frame, the motion model described above was used to compute a final score $p(S_k) = o_w(S_k) + \alpha_m m(S_k)$ for each segment S_k in the frame t. The highest $p(S_k)$ from tracking is then compared with the stored p_t for this track from the inference step. If $\max_k p(S_k) > p_t$, then segment S_k is selected as the representative. Otherwise, the segment that came from the inference step is selected. This procedure is run until the end of the sequence. This procedure can significantly improve the appearance model and lead to improved tracking performance.

4.4 Deep Matting Refinement

The segment masks generated with this approach may not adhere well to boundaries. Inspired by Xu in (Xu et al., 2017), Shaban et al., 2017a construct a mask refinement network similar

to an image matting network. We briefly review the proposed method in (Shaban et al., 2017a) in this section.

In image matting, the user specifies a trimap consisting of areas that must/cannot belong to the object and areas that may or may not belong to the object. Then, a refinement deep network takes both the image and trimap as inputs and outputs an improved segment.

Instead of the trimap, Shaban et al., 2017a use the pixel-level confidence map from the SPT tracking algorithm. For each segment track, denote its weight w_i to be its predicted overlap $o_w(G_0)$ on the ground truth segment G_0 in the first frame, if G_0 is given by the user. Then, all the tracks with $w_i > 0.7 \max_i w_i$ are weighted-averaged in each frame to create a pixel-level confidence map (Fig.4.6). This is utilized instead of the trimap to train a network similar in (Xu et al., 2017) with a few differences in the details: First, since our prediction is a binary mask rather than alpha channel, we removed composition loss. Second, we squared the loss function in (Xu et al., 2017), which resulted in better performance in practice. Third, we removed the skip path in the Matting Refinement Stage. Finally, we set a threshold of 128 to binarize our output image.

The Matting Encoder-Decoder stage is pretrained on the matting dataset in (Xu et al., 2017), and then fine-tuned on the composition of DAVIS dataset objects as foreground and the COCO images as background. During testing, this network significantly improves the boundary coherence of the tracked segments.

4.5 Experiments

We test different settings with different combinations of sequence-independent proposals and sequence-specific proposals. First, a semi-supervised setting in which all the target objects are fully annotated in the first frame **PGN+FCIS**. Second, when only m objects are annotated in the first frame **PGN$_m$+FCIS**. For this purpose we only use the annotations for the objects in which their best IoU with FCIS proposals are less than 75% in the first frame. This simulates the interactive process in which the user sees the unsupervised results with

FCIS proposals in the first frame, select/rejects proposed tracks, and provides annotations for the missed objects. Finally, we test separately the scenarios using only **FCIS** proposals (only semantic information, no ground truth annotation), only using **POISE** proposals (completely unsupervised) and using **FCIS + POISE** proposals (no interactive annotations). $+$ in the superscript shows the result after applying deep mating refinement step. More experiments on performance of the combinatorial grouping and parameter tuning for FCIS could be found in the supplementary materials. Per-sequence results are also reported in supplementary material.

We measured the performance of our algorithm on three different benchmarks (DAVIS-2016, DAVIS-2017 and SegTrack v2), reporting the standard IoU (intersection-over-union) measure averaged over all instances and all frames (excluding the first frame which was annotated) in each video. We run all the experiments on a machine with 4 CPU cores and a Titan Xp GPU. Since we are running the tracking code on CPUs, our runtime is not directly comparable to those methods that depend on GPUs. Altogether the algorithm takes about 3-6 seconds per frame. Most of the time is spent on computing the motion term. We believe with further code optimization and GPU parallelization this algorithm can be made close to real-time.

Table 4.1: Video object segmentation results on DAVIS-2017 test-challenge set.

	Global	**Region** $\mathcal{J}$			**Boundary** $\mathcal{F}$		
Team Name	Mean	Mean	Recall	Decay	Mean	Recall	Decay
lixx (Li et al., 2017)	0.699	0.679	0.746	0.225	0.719	0.791	0.241
apta (Khoreva et al., 2017)	0.678	0.651	0.725	0.277	0.706	0.798	0.302
vantam299 (Le et al., 2017)	0.638	0.615	0.686	0.171	0.662	0.79	0.176
PGN+FCIS	0.615	0.598	0.710	0.219	0.632	0.746	0.237
voigtlaender (Voigtlaender and Leibe, 2017)	0.577	0.548	0.608	0.31	0.605	0.672	0.347
cjc (Cheng et al., 2017)	0.569	0.536	0.595	0.253	0.602	0.679	0.276
Fromandtozh (Zhao, 2017)	0.539	0.507	0.549	0.325	0.571	0.632	0.337
anewswan (Newswanger and Xu, 2017)	0.509	0.490	0.551	0.213	0.528	0.583	0.237
ilanv (Sharir, Smolyansky, and Friedman, 2017)	0.497	0.460	0.493	0.331	0.533	0.584	0.364

Table 4.2: Video object segmentation results on DAVIS-2017 test-dev set.

Team Name	Global Mean	Region $\mathcal{J}$ Mean	Region $\mathcal{J}$ Recall	Region $\mathcal{J}$ Decay	Boundary $\mathcal{F}$ Mean	Boundary $\mathcal{F}$ Recall	Boundary $\mathcal{F}$ Decay
apata (Khoreva et al., 2017)	0.666	0.624	0.733	0.193	0.707	0.814	0.197
lixx (Li et al., 2017)	0.661	0.644	0.735	0.245	0.678	0.756	0.271
PGN+FCIS	0.576	0.545	0.635	0.139	0.608	0.706	0.144
voigtlaender (Voigtlaender and Leibe, 2017)	0.565	0.534	0.578	0.199	0.596	0.654	0.190
ilanv (Sharir, Smolyansky, and Friedman, 2017)	0.558	0.519	0.557	0.176	0.598	0.658	0.189
Fromandtozh (Zhao, 2017)	0.552	0.524	0.584	0.181	0.579	0.661	0.200

Table 4.3: Ablation results on DAVIS-2017 test-dev.

Method	Global Mean	Region $\mathcal{J}$ Mean	Region $\mathcal{J}$ Recall	Region $\mathcal{J}$ Decay	Boundary $\mathcal{F}$ Mean	Boundary $\mathcal{F}$ Recall	Boundary $\mathcal{F}$ Decay
Semi-Supervised							
PGN+FCIS	57.6	54.5	63.5	13.9	60.8	70.6	14.4
PGN$_{38}$+FCIS	56.8	54.3	65.4	13.9	59.4	68.7	15.6
SPT Only	54.5	51.6	61.2	16.6	57.4	66.7	16.1
Semantic and Unsupervised							
FCIS	37.6	36.3	42.9	8.0	39.0	43.2	8.5
POISE	32.6	31.5	30.5	17.8	33.7	30.0	18.8

4.5.1 DAVIS-2017

We evaluate the performance of our algorithm on the `test-challenge` and `test-dev` set in DAVIS-2017 (Pont-Tuset et al., 2017). As other methods are fully semi-supervised we only include the results of our method when all the instances are labeled in the first frame. We later study the effect of decreasing the number of labeled instances in the first frame. For the `test-challenge` we only used a simple dense CRF (Krähenbühl and Koltun, 2011) for the mask refinement as the deep mating network was not ready at the time of the competition, and no submission was allowed after the challenge. The improvement that comes from the dense CRF approach was 0.9%. Results for the `test-challenge` set are shown in Table 4.1. The top 3 methods use the mask propagation idea. lixx (Li et al., 2017) uses Resnet-101 (He et al., 2016) trained on COCO dataset specifically for mask propagation. LucidDream (apta) (Khoreva et al., 2017) is the ensemble of 4 variation of mask propagation, each are fine-tuned on the first frame annotations for 40k iterations. Our algorithm significantly outperforms online OSVOS (Voigtlaender and Leibe, 2017)

Methods	NLC (Faktor and Irani, 2014)	FSEG (Jain, Xiong, and Grauman, 2017)	ARP (Koh and Kim, 2017)	FCP (Perazzi et al., 2015)	BVS (Märki et al., 2016)	OSVOS (Caelles et al., 2017a)	MSK (Perazzi et al., 2017)	PGN*	PGN+FCIS	PGN+FCIS*	PGN_t+FCIS	PGN_t+FCIS*
$\mathcal{J}$	0.551	0.707	0.762	0.584	0.600	0.798	0.797	0.771	0.814	**0.843**	0.819	0.842

Table 4.4: Video object segmentation results on DAVIS-2016 val set. Our algorithm greatly outperforms unsupervised (first three methods) and semi-supervised algorithm (next four methods) with only having access to full annotation of 4 objects out of 20.

(voigtlaender) and improved version of OSVOS (Zhao, 2017) (Fromandtozh). The results from (anewswan) come from a slight improvement of OSVOS, hence it could be deducted that the method improved at least 11% on top of OSVOS in this dataset. The first and the third places used special treatment for persons to improve performance, which is not the case for our model. The results on `test-dev` set are also reported in Table 4.2. The application of these listed methods are limited to the semi-supervised setting where annotation for all instances are available in the first frame, while ours can track many more objects without any annotation. Some qualitative results are shown in Fig. 4.7.

Ablation Study First, we analyze the effect of annotating less objects in the first frame. We compare PGN+FCIS with PGN_{38}+FCIS, in which only 38 objects out of 89 objects (less than 40%) are annotated. In PGN_{38}+FCIS, we only use the annotations for the objects in which their best IoU with FCIS proposals are less than 65% in the first frame. Our algorithm shows similar performance by having access to only 40% of the annotations. We also show ablation results for SPT+the first backtracking step (without it many sequences won't even have a track starting from the 1st frame), denoted as **SPT Only**. This is about 3% worse than SPT-Retrack on DAVIS-2017 `test-dev`. Finally, performance without the first-frame annotations are shown, with **FCIS**, **POISE**. FCIS is incapable of tracking many objects that are not similar to those in the COCO dataset. POSIE returns many track proposals for each object and most of the performance drop happens in choosing the best track by testing them over only the first frame annotations.

DAVIS-2016 (Perazzi et al., 2016) We evaluated our algorithm on the validation set with

Table 4.5: Video object segmentation results on SegTrack v2.

Methods	TRS (Xiao and Jae Lee, 2016)	BVS (Märki et al., 2016)	OSVOS (Caelles et al., 2017a)	MSK (Perazzi et al., 2017)	ObjFlow (Tsai, Yang, and Black, 2016)	**PGN+FCIS**
$\mathcal{J}$	0.691	0.584	0.650	0.703	0.765	**0.735**

Figure 4.7: Qualitative results from the Algorithm. The algorithm handles changing appearances and occlusions well

20 sequences (1,376 frames). There is a single foreground target in each sequence and this dataset is generally significantly simpler than DAVIS-2017. We have summarized the overall results in Table 4.4. Please see the supplementary material for the detailed per-sequence results. Using FCIS proposals and SPT with PGN improves the results by 4.3%. Surprisingly, using PGN proposals only for the 4 instances out of 20 leads to better segmentation results. This shows that using semi-supervised proposals for the objects that FCIS can detect is not only unnecessary, but can even hurt the performance by introducing low-quality proposals to the tracking algorithm. The deep refinement network increases the performance by 3% in both cases. SPT beats OSVOS and mask propagation method MSK that do not utilize a long-term tracking model.

SegTrackv2 SegTrack-V2 is a benchmark dataset for multiple object segmentation. It has a total of 14 sequences with 24 instances. The evaluation results are listed in Table 4.5. Our method outperforms all the listed algorithms and reaches the state-of-the-art on this dataset. It is important to note that FCIS in this dataset generalizes well to instances like penguin, parachute, monkey, cheetah, and frog, which are not present in COCO.

4.6 Conclusion

This paper proposes a framework that accommodates unsupervised, semi-super-vised and semantic video segmentation in the same video segmentation system. This would be useful in videos where many objects come from known categories, hence without any annotation the system can already segment and track them, while still accommodating the cases where additional segments need to be annotated and tracked. An improved version of SPT is proposed to jointly track object proposals from different sources. We also proposed to use deep image matting to improve the boundary coherence of the tracked segments. Experiments showed that our algorithm works well even when only a few objects are annotated, thus broadly applicable to different use cases of video segmentation.

CHAPTER 5

TOWARDS FEW-SHOT WEAKLY SUPERVISED OBJECT DETECTION

Few-shot semantic segmentation problem discussed in Chapter 3 is an instance of supervised few-shot learning since the target object is fully annotated in the support set. In this chapter, we introduce few-shot weakly supervised detection problem where only image level labels of images in the support set are available. Given the weakly supervised support set, the task is to detect the target object in an query image. For this purpose, we first address the problem of finding images of a common object across bags of images. In Section 5.4 we and use this algorithm to annotate the images in the support set. Once the support set is fully annotated, any off-the-shelf supervised few-shot detection algorithm can be used to detect these objects in the query image.

The problem of finding common object across multiple images is a fundamental task in compute vision by itself. In addition to the few-shot weakly supervised object detection problem, several other problems, including co-segmentation, co-localization, and unsupervised video object tracking and segmentation have been also formulated in this way (Babenko, Yang, and Belongie, 2009; Faktor and Irani, 2013; Fu et al., 2014; Hsu et al., 2018; Vicente, Rother, and Kolmogorov, 2011). We spend the first part of this chapter to addressing this problem. In the next parts, we discuss the application of the proposed method for large scale weakly supervised object localization and few-shot weakly supervised object detection.

Finding common objects across few image collections

The input is a collection of bags, each containing several images from multiple classes. A bag is labelled as *positive* with respect to a given object class if it contains at least one image from that class and *negative* if none of the images in the bag are from the object class. The task is to find an instance of the common object in each positive bag. It is not assumed that

Figure 5.1: Co-localization, shown here, is an instance of the general problem of finding common objects addressed in this chapter. Each image in the top row generates a positive bag containing a set of cropped regions from that image. The task is to find a common object from the positive bags by selecting one region from each image (green bounding boxes). Cropped regions from the images in the bottom row form a negative bag as they do not contain the common object. The negative bag is optional here but can reduce ambiguity. For example, since a knife is present in the negative bag it can not be the desired common object.

objects of the common class have been seen previously during training.

Since collections of images may accidentally contain irrelevant common objects (for instance indoor images often contain person), the purpose of a negative bag is to indicate objects we are *not* looking for, but which may be common to the positive bags.

Several computer vision problems, including co-localization can be formulated in this way. In the co-localization problem, Figure 5.1, each bag contains many cropped image regions (object proposals) from one image. The goal is to identify proposals, one per positive bag, that contain the common object. We design our approach to address the general problem of finding common objects from positive bags and evaluate it on two problems: few-shot common object recognition and object co-localization. We show applications of

this approach in large scale weakly supervised object localization and few-shot weakly supervised object detection in Section 5.3 and 5.4 respectively.

Weakly supervised classification methods like multiple-instance learning (Maron and Lozano-Pérez, 1998) have been used to address this type of problems, but they require many training bags to learn new concepts (Ilse, Tomczak, and Welling, 2018). Meta-learning techniques (Finn, Abbeel, and Levine, 2017b; Santoro et al., 2016; Shaban et al., 2019b) have been shown to reduce the need for training instances in few-shot learning, but these methods require full supervision for the new classes.

We model the problem of finding common objects as a minimum-energy graph labelling problem, otherwise known as a bidirectional graphical model or Markov Random Field. Each node of the graph corresponds to a positive bag and a graph labelling corresponds to choosing one image in each positive bag, the goal being to find a labelling that contains the common object. We use the word *selection* instead of *labelling* to refer to the process of selecting one image from each bag. The energy minimization problem uses unary and pairwise potential functions, where unary potentials reflect the relation of images in the positive bags to the images in the negative bag and the pairwise potentials derive from a similarity measure between pairs of images from two positive bags. The unary and pairwise potentials are computed using similar, but separately trained networks. We adapt the relation network (Sung et al., 2018), which has been successfully used in few-shot recognition to compute pairwise potentials, and propose a new algorithm that uses the relationship of an image to all of the images in the negative bag to provide unary potentials.

Once unary and pairwise potentials have been computed, any off-the-shelf inference algorithm can be used to find a minimum-cost labelling. In our experiments, we compare several optimization algorithm and discuss their advantages for solving our problem.graph labelling.

Although graphical models have been used for Multiple Instance Learning (MIL) problems (Deselaers and Ferrari, 2010; Hajimirsadeghi et al., 2013), our method uses a learning-

based approach, inspired by meta-learning, to increase the generalization power of potential functions to novel classes.

Few-Shot Weakly Supervised Object Detection

The problem setup for few-shot weakly supervised task is very similar to few-shot image segmentation in Chapter 3. In this problem, the goal is to provide bounding box annotation for objects' instances rather than pixel level annotation. Unlike the supervised counterpart, we do not have access to the object annotations in the support set. Instead, labels only show foreground classes that are present in each the image of the support set. The goal is to learn a model that, when given support set, predicts label of all bounding box proposals in a query image. We solve the problem of few-shot weakly supervised object detection in a two stage process. In the first stage, we use the proposed co-localization method to annotate the support set images. The proposed algorithm for finding common object only selects one image from each bag. Using only the top selection may not be optimal due to the ambiguities in finding the correct common object across few bags. To increase the flexibility of the algorithm, we utilize M-best Mode method in (Batra et al., 2012) to extract top M qualitatively distinct selections. Finally, we propose a simple few-shot detection algorithm that receives the annotated support set, learns objects representations, and classifies the bounding box proposals in the query image.

Contributions

Overall, we make the following contributions:

1. We propose a method for finding common objects across few image collections in Section 5.2. Our approach transfers knowledge from large scale strongly supervised datasets and utilizes this knowledge to localize previously unseen objects in new images. We demonstrate the superiority of this learned relation metric to earlier MIL approaches on co-localization problem.

2. In Section 5.3, we further show the effectiveness of the proposed knowledge transfer method in improving the performance of large scale object localization.

3. In Section 5.4, we extend the proposed method to find multiple common object proposals across image collections and study its application to few-shot weakly supervised object detection.

5.1 Related Work

Multiple instance learning (MIL) (Carbonneau et al., 2018; Pathak et al., 2015) methods have been used for learning weakly supervised tasks such as object localization (WSOL) (Chen et al., 2017; Jie et al., 2017; Shen et al., 2018; Zhuang et al., 2017). In a standard MIL framework, instance labels in each positive bag are treated as hidden variables with the constraint that at least one of them should be positive. MI-SVM and mi-SVM (Andrews, Tsochantaridis, and Hofmann, 2003) are two popular methods for MIL, and have been widely adapted for many weakly supervised computer vision problems, achieving state-of-the-art results in many different applications (Carbonneau et al., 2018; Doran and Ray, 2014). In these methods, images in each bag inherit the label of the bag and an SVM is trained to classify images. The trained SVM is used to relabel the instances and this process is repeated until the labels remain stable. While in MI-SVM only the image with the highest score in positive bags are labeled as positive, mi-SVM allows more than one positive label in each positive bag in the relabeling process.

Co-saliency (Hsu et al., 2018; Zhang et al., 2015), co-segmentation (Faktor and Irani, 2013; Hochbaum and Singh, 2009; Vicente, Rother, and Kolmogorov, 2011), and co-localization (Li et al., 2016a) methods have the same kind of output as WSOL methods. Similar to standard MIL algorithms, some of these methods rely on a relatively large training set for learning novel classes (Li et al., 2016a; Tang et al., 2014). The main difference between these methods and WSOL methods is that they usually do not utilize negative examples (Li et al., 2016a; Tang et al., 2014; Vicente, Rother, and Kolmogorov, 2011).

Negative examples in our method are optional and could be used to improve the results of the co-localization task.

Our approach is related to weakly supervised methods that make use of auxiliary fully-labelled data to accelerate the learning of new categories (Deselaers, Alexe, and Ferrari, 2012; Hoffman et al., 2015; Rochan and Wang, 2015; Shi, Caesar, and Ferrari, 2017; Uijlings, Popov, and Ferrari, 2018). Since visual classes share many visual characteristics, knowledge from fully-labelled source classes is used to learn from the weakly-labelled target classes. The general approach is to use the labelled dataset to learn an embedding function for images and use MI-SVM to classify instances of the weakly labelled dataset in this space (Hoffman et al., 2015; Shi, Caesar, and Ferrari, 2017; Uijlings, Popov, and Ferrari, 2018). We show that learning a scoring function to compare images in the embedded space significantly improves the performance of this approach, especially when few positive images are available. Rochan et al. (Rochan and Wang, 2015) propose a method to transfer knowledge from a set of familiar objects to localize new objects in a collection of weakly supervised images. Their method uses semantic information encoded in word vectors for knowledge transfer. In contrast, our method uses the similarity between tasks in training and testing and does not rely solely on a given semantic relationship between the familiar and new classes. Deselaers et al. (Deselaers, Alexe, and Ferrari, 2012) transfer objectness scores from source classes and incorporate them into unary terms of a conditional random field formulation.

Our approach inspired by methods that use the meta-learning paradigm for few-shot classification. These methods simulate the few-shot learning task during the training phase in which the model learns to optimize over a batch of sampled tasks. The meta-learned method is later used to optimize over similar tasks during testing. Optimization-based methods (Finn, Abbeel, and Levine, 2017b; Ravi and Larochelle, 2017b), feature and metric learning methods (Snell, Swersky, and Zemel, 2017; Sung et al., 2018; Vinyals et al., 2016), and memory augmented-based methods (Santoro et al., 2016) are just a few

examples of modern few-shot learning. While our work is inspired by these methods, it is different in the sense that we do not assume strong supervision for the tasks. In relation networks (Sung et al., 2018) a similarity function is learned between image pairs and used to classify images from unseen classes. We adopt this method to learn the unary and pairwise potential functions in our graphical model.

5.2 Finding Common Object Across Few Image Collections

5.2.1 Problem description

We consider a set $\mathcal{I}$ with a binary relation R. The elements of the set are called *images* in our work for simplicity of exposition. A *relation* R is simply a subset of $\mathcal{I} \times \mathcal{I}$:

$$R(e, e') = \begin{cases} +1, & \text{if } (e, e') \in R \text{ (inputs are related)} \\ -1, & \text{otherwise.} \end{cases} \tag{5.1}$$

A *bag* is a set of images, thus, a subset of $\mathcal{I}$. We will be concerned with collections of bags, $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$. We say that a collection $\mathcal{V} = \{v_1, \ldots, v_N\}$ is *consistent* if it is possible to select images, one from each bag, so that they are all related in pairs. These are known as *positive* bags.

Given a consistent collection, $\mathcal{V}$ and an optional additional bag $\bar{v}$ that we designate as *negative*[1], the task is to output a *selection* of images, namely an ordered set $O = (e_1, \ldots, e_N)$ where e_i is from positive bag v_i, such that the images are pairwise related, $R(e_i, e_j) = 1$, and that not all images are pairwise related to any image in the negative bag, i.e., $\exists e_i \in O$ such that $\forall \bar{e} \in \bar{v}, R(e_i, \bar{e}) = -1$.

The situation of most interest is where each of the images $c_e \in \{c_\varnothing\} \cup \mathscr{C}$ where $c_\varnothing$ is a *background class* and $\mathscr{C}$ is a set of *foreground classes*. Two images e_1 and e_2 are related if their labels are the same and belong to a foreground class, i.e., $c_{e_1} = c_{e_2} \in \mathscr{C}$. For example,

[1]There is no point in having more than one negative bag in a collection since its purpose is simply to provide a set of images that are not compatible with the positive bags, in the sense described.

(cropped) images may be labelled according to the foreground object they contain. In this case, two images (e_1, e_2), both containing a "cake" are related, $R(e_1, e_2) = 1$. Whereas two images (e_3, e_4) that are not of the same foreground class are unrelated, $R(e_3, e_4) = -1$. In this case, R is an equivalence relation.

Energy function. We pose the problem of finding the common object as finding a selection O that minimizes an energy function. Our energy function is defined as sum of potential functions as follows:

$$E(O \mid \bar{v}) = \sum_{\substack{e_i, e_j \in O \\ i>j}} \psi_\theta^{\mathrm{P}}(e_i, e_j) + \eta \sum_{e_i \in O} \psi_\beta^{\mathrm{U}}(e_i \mid \bar{v}), \tag{5.2}$$

in which $\psi_\theta^{\mathrm{P}}(\cdot, \cdot)$ and $\psi_\beta^{\mathrm{U}}(\cdot \mid \bar{v})$ are pairwise and unary potential functions with trained parameters θ and β, and hyperparameter $\eta \geq 0$ controls the importance of the unary terms. Both unary and pairwise potential functions are learned by neural networks, which will be described in Section 5.2.3. The pairwise potential function is learned so that it encourages choosing pairs that are related to each other. The unary potential is chosen so it is minimized when its input is not related to the images in the negative bag. In this way, the overall energy is minimized when images in O are related to each other and unrelated to images in the negative bag.

5.2.2 Training and Test Splits

For a dataset $\mathcal{D} \subseteq \mathcal{I}$, we use the notation $\mathcal{W} \sim \mathcal{D}$ to indicate that a random collection $\mathcal{W} = (\mathcal{V}, \bar{v})$ is drawn from the dataset. We define the sampling strategy in the implementation details for each dataset. During training, algorithms have access to a dataset $\mathcal{D}_{\text{train}}$ and corresponding ground-truth relation. We construct the relation for the training dataset based on a set of foreground classes $\mathscr{C}_{\text{train}}$ as described above.

Methods are evaluated on samples from a test dataset $\mathcal{W} \sim \mathcal{D}_{\text{test}}$. There are no image in

common between the training and test datasets. Moreover, the set of foreground classes $\mathscr{C}_{\text{test}}$ used for the test dataset is disjoint from the set of foreground classes used during training, i.e., $\mathscr{C}_{\text{test}} \cap \mathscr{C}_{\text{train}} = \varnothing$. At test time we only know whether a bag is positive or negative with respect to some foreground class. The ground-truth (which foreground class is common to the positive bags) is unknown to the algorithm and is only used for evaluating performance.

5.2.3 Learning the potential functions

We now present the method for learning the pairwise and unary potential functions. The proposed method relies on an algorithm to estimate a similarity measure of an input image pair (e, e'). One common approach is to learn an embedding function and use a fixed distance metric to compare the input pairs in the embedded space. In this approach, the learning is used only to determine the embedding function. The relation network (Sung et al., 2018) extends this by jointly learning the embedding function and a comparator. The network consists of embedding and relation modules. The *embedding module* learns a joint feature embedding (into $\mathbb{R}^d$) for the input pair of images $\mathcal{C}(e, e')$ and the *relation module* learns a mapping $g : \mathbb{R}^d \to \mathbb{R}$, mapping the embedded feature to a relation score $r_\phi(e, e') = g(\mathcal{C}(e, e'))$ where ϕ denotes the parameters of the embedding and scoring functions combined.[2] We adopt the relation module from the Relation Network due to its simplicity and success in few-shot learning. However, any other method which computes the relationship between a pair of images could be used in our method.

Relation network. As we need to evaluate the relation of many image pairs, we adapt the original relation network architecture (Sung et al., 2018) in order to make the embedding and scoring functions as computationally efficient as possible. The feature embedding function $\mathcal{C}(\cdot, \cdot) : \mathcal{I} \times \mathcal{I} \to \mathbb{R}^d$ consists of feature concatenation and a single linear layer with gated activation (Oord et al., 2016) and skip connections. Let f and f' be features in $\mathbb{R}^d$ extracted from images e and e' by a CNN *feature extraction module* and $[f, f']$ be the concatenation

[2] We adopt the notation used in the relation network paper (Sung et al., 2018)

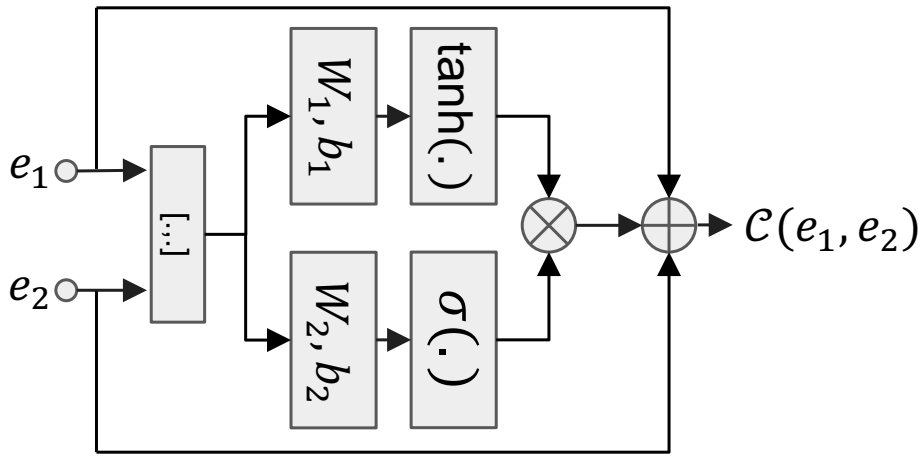


Figure 5.2: *Feature Embedding Module $\mathcal{C}(.,.)$. Input feature pairs are embedded into a joint embedding function by a gated activation layer.*

of feature pairs. The embedding function, shown in Figure 5.2 is defined as:

$$\mathcal{C}(e, e') = \tanh(W_1[f, f'] + b_1)\sigma(W_2[f, f'] + b_2) + \frac{f + f'}{2}$$

where $W_1, W_2 \in \mathbb{R}^{d \times 2d}$ and vectors $b_1, b_2 \in \mathbb{R}^d$ are the parameters of the feature embedding module and $\tanh(\cdot)$ and $\sigma(\cdot)$ are hyperbolic tangent and sigmoid activation functions respectively, applied componentwise to vectors in $\mathbb{R}^d$.

Then, we use a linear layer to map this features into relation score

$$r_\phi(e, e') = w^\top \mathcal{C}(e, e') + b$$

where $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$. We found in practice that using gated activation in the embedding module improves the performance over a simple ReLU, whereas adding more layers does not affect the performance. We note that the effectiveness of gated activation has also been shown in other work (Ramachandran, Zoph, and Le, 2017).

Pairwise potentials. The pairwise potential function is defined as the negative of the output of the relation module: $\psi^{\mathrm{P}}_\theta(e_i, e_j) = -r_\theta(e_i, e_j)$ so it has a lower energy for related

pairs. For a sampled collection $\mathcal{V}$ the episode loss is written as a binary logistic regression loss

$$\mathcal{L}^{\mathrm{P}} = \frac{1}{N_P} \sum_{(e_i,e_j)\sim\mathcal{V}} \log\Big(1 + \exp(-R(e_i,e_j) r_\theta(e_i,e_j))\Big)$$

where the sum is over all the pairs in the collection, N_P is the total number of such pairs, and relation $R(.,.)$ defined in Eq (5.1) provides the ground-truth labels.

Note that image pairs are sampled from $\mathcal{V}$, so that the loss function reflects prior distributions of image pairs from consistent image collections.

Unary potentials. We describe the method proposed in (Shaban et al., 2019a) to efficiently predict the unary potentials. The unary potential $\psi^{\mathrm{U}}(e \mid \bar{v})$ is constructed by comparing image e with images in the negative bag $\bar{v}$. Let the vector $u(e,\bar{v})$ be the estimated relation between image e and all the images in $\bar{v}$, that is, $u(e,\bar{v})_j = r_\beta(e,\bar{e}_j)$ where $\bar{e}_j$ is the j-th image in the negative bag and β is the (new) set of parameters for the relation network. By definition, the unary energy for an image e should be high if at least one of the values in $u(e,\bar{v})$ is high. In other words, e is related to $\bar{v}$ if it is related to at least one image in $\bar{v}$. This suggests the use of $\max_j(u(e,\bar{v})_j)$ as the unary energy potential. However, depending on the class distribution of images in the negative bag, an image e which is not from the common object class could be related to more than just one image from the negative bag. In this case, using the average relation to the few mostly related elements in $u(e,\bar{v})$ helps to reduce the noise in the estimation and works better than a simple max operator. This motivates us to use a form of exponential-weighted average of the relations so that higher values get a higher weight

$$\psi^{\mathrm{U}}_{\beta,\nu}(e \mid \bar{v}) = \frac{\sum_{k=1}^{\bar{B}} u(e,\bar{v})_k \exp\big(\nu\, u(e,\bar{v})_k\big)}{\sum_{k=1}^{\bar{B}} \exp(\nu\, u(e,\bar{v})_k)}\,. \tag{5.3}$$

Here, $\bar{B}$ is the total number of images in the negative bag and ν is the temperature parameter. Observe that for $\nu = 0$ we have the mean value of $u(e,\bar{v})$ and it converges to the max

operator as $\nu \rightarrow +\infty$. We let the algorithm learn a balanced temperature value in a data-driven way.

For a sampled collection $\mathcal{W} = (\mathcal{V}, \bar{v})$, the episode loss for the unary potential is defined as a binary logistic regression loss

$$\mathcal{L}^{\mathrm{U}} = \frac{1}{N_U} \sum_{v \in \mathcal{V}} \sum_{e \in v} \log \Big(1 + \exp(-R(e, \bar{v}) \psi^{\mathrm{U}}_{\beta,\nu}(e, \bar{v})) \Big) \tag{5.4}$$

where we use an extended definition of the relation function where $R(e, \bar{v}) = \max_{\bar{e} \in \bar{v}} R(e, \bar{e})$ and N_U is the total number of images in all positive bags in the collection. Through training, this loss is minimized over choices of parameters β of the relation network, and the weight parameter ν. By optimizing this loss, we learn a potential function that has higher value if e is related to one example in the negative bag. Note that in Eq (5.2) selection of unary potentials with high values are discouraged.

As before, training samples are chosen from collections $\mathcal{W}$ to reflect the prior distributions of related and unrelated pairs.

Parameters of the unary and pairwise potential functions are learned separately by optimizing the respective loss functions over randomly sampled problems from the training set. Although both unary and pairwise potential functions use the relation network with an identical architecture, their input class distributions are different, since one is comparing images in positive bags and one is comparing images in positive and negative bags. Thus, sharing their parameters decreases overall performance.

5.2.4 Inference

Finding an optimal selection O that minimizes the energy function defined in Eq (5.2) is NP-hard and thus not feasible to compute exactly, except in small cases. Loopy belief propagation (Weiss and Freeman, 2001), TRWS (Kolmogorov, 2006), and AStar (Bergtholdt et al., 2010), are among well-known algorithms used for approximate energy minimization.

In our experiments, we use the state-of-the-art TRWS algorithm for inference.

5.2.5 Experiments

We evaluate the proposed algorithm on few-shot common object recognition and co-localization tasks. For each task, we first pre-train a CNN feature extractor module to perform classification on the seen categories from the training dataset. We then use the learned CNN to compute a feature descriptor of each image. This ensures a consistent image representation for all methods under consideration.

For learning pairwise and unary potentials, stochastic gradient descent with gradual learning rate decay schedule is used. The complete framework ("Ours" in the tables) uses TRWS optimization method described in (Kolmogorov, 2006) for inference. We use a highly efficient parallel implementation of this algorithm (Andres, Beier, and Kappes, 2012). The optimal value of η in Eq (5.2) is found using grid search. In all experiments, a maximum of $k = 300$ top selection proposals are kept in the greedy algorithm.

All experiments are done on a single Nvidia GTX 2080 GPU and 4GHz AMD Ryzen Threadripper 1920X CPU with 12 Cores[3].

5.2.6 Baseline Methods

The proposed method is compared to SVM based and attention based MIL baselines described below.

SVM based MIL. We report the results of the three well-known approaches: MI-SVM (Hoffman et al., 2015), mi-SVM (Andrews, Tsochantaridis, and Hofmann, 2003) and sbMIL (Bunescu and Mooney, 2007) using publicly available source code (Doran and Ray, 2014). The sbMIL method is specially designed to deal with sparse positive bags. The RBF and linear kernel are chosen as they work better on few-shot common object recognition and co-localization respectively. Grid search is performed in order to select the hyperparameters.

[3]The code is publicly available https://github.com/haamoon/finding_common_object.

Attention based deep MIL. Along with the SVM based methods, the results of the more recent attention based deep learning MIL method (Ilse, Tomczak, and Welling, 2018) (ATNMIL) is presented on our benchmarks. After training the model, we select the image proposal with the maximum attention weight from each positive bag.

5.2.7 Few-shot Common Object Recognition

In this task we make use of the *mini*ImageNet dataset (Vinyals et al., 2016). The advantage of *mini*ImageNet is that we can compare many different design choices without requiring large scale training and performance evaluations. The dataset contains $60,000$ images of size 84×84 from 100 classes. We experiment on the standard split for this task of 64, 16 and 20 classes for training, validation and testing, respectively (Ravi and Larochelle, 2017b).

For the CNN feature extractor module, a Wide Residual Network (WRN) (Zagoruyko and Komodakis, 2016) with depth 28 and width factor 10 is pre-trained on the training split. The $d = 640$ dimensional output of global average pooling layer of the pre-trained network is provided as input to all the methods.

To construct bags, we first randomly select M classes out of all the possible classes $\mathscr{C}$. One of these is selected to be the target and the rest are considered non-target classes. Then, each positive bag is constructed by randomly sampling one image from the target class and $B - 1$ images from the target and non-target classes. The negative bag is built by sampling $\bar{B}$ examples from non-target classes. For output selection O, we measure the *success rate* which is equal to the percentage of $e \in O$ that belong to the target class. We compute the expected value of success rate for 1000 randomly sampled problems and report the mean and 95% confidence interval of the evaluation metric.

We vary the number of bags as well as their sizes. We select the number of positive bags $N \in \{4, 8, 16\}$, the size of each positive bag $B \in \{5, 10\}$, and the size of negative bag $\bar{B} \in \{10, 20\}$. The number of classes M to sample from in each episode changes the difficulty of the task. Lower values of M make the problem more ambiguous by increasing

Table 5.1: Success rate on *mini*ImageNet for different positive bags N, and total number of negative images $\bar{B}$. The first and the second part of the table show the results for bag size 5 and 10 respectively.

	N	4		8		16	
	$\bar{B}$	10	20	10	20	10	20
Bag Size = 5	Ours	$\mathbf{63.78 \pm 1.49}$	$\mathbf{65.43 \pm 1.47}$	$\mathbf{72.60 \pm 0.98}$	$\mathbf{73.80 \pm 0.96}$	$\mathbf{78.71 \pm 0.63}$	$\mathbf{80.08 \pm 0.62}$
	Baseline	60.88 ± 1.51	63.83 ± 1.49	64.46 ± 1.05	68.08 ± 1.02	66.78 ± 0.73	70.39 ± 0.77
	MI-SVM	56.25 ± 1.54	59.03 ± 1.52	62.75 ± 1.06	63.76 ± 1.05	67.91 ± 0.72	73.33 ± 0.69
	sbMIL	54.55 ± 1.54	59.93 ± 1.52	58.25 ± 1.08	64.68 ± 1.05	61.35 ± 0.75	65.55 ± 0.74
	mi-SVM	54.23 ± 1.54	59.43 ± 1.52	60.43 ± 1.07	66.08 ± 1.04	64.49 ± 0.74	69.69 ± 0.71
	ATNMIL	50.35 ± 1.55	60.33 ± 1.52	56.05 ± 1.09	63.29 ± 1.06	58.97 ± 0.76	67.26 ± 0.73
Bag Size = 10	Ours	$\mathbf{37.15 \pm 1.50}$	38.50 ± 1.51	$\mathbf{42.61 \pm 1.08}$	$\mathbf{47.59 \pm 1.09}$	$\mathbf{50.88 \pm 0.77}$	$\mathbf{53.71 \pm 0.77}$
	Baseline	35.73 ± 1.49	$\mathbf{40.40 \pm 1.52}$	38.01 ± 1.06	43.95 ± 1.09	41.08 ± 0.76	47.83 ± 0.77
	MI-SVM	29.53 ± 1.41	35.05 ± 1.48	35.25 ± 1.05	39.94 ± 1.07	41.21 ± 0.76	46.63 ± 0.77
	sbMIL	31.55 ± 1.44	31.50 ± 1.44	34.10 ± 1.04	39.86 ± 1.07	28.80 ± 0.70	43.63 ± 0.77
	mi-SVM	31.55 ± 1.44	35.33 ± 1.48	34.10 ± 1.04	39.86 ± 1.07	39.48 ± 0.76	45.16 ± 0.77
	ATNMIL	26.58 ± 1.37	33.10 ± 1.46	28.48 ± 0.99	35.11 ± 1.05	31.56 ± 0.72	38.14 ± 0.75

the chance of generating other common objects in subproblems. On the other hand, it increases the importance of the negative bag by increasing the chance of having more samples from each non-target class. We randomly choose M between 5 and 15 when $B = 5$, and between 10 and 20 when $B = 10$ for each problem.

The results in Table 5.1 show our method outperforms ATNMIL and SVM based approaches for all versions of the problem. To test the importance of learning the unary and pairwise potentials, we construct a baseline that uses cosine similarity to compute the relation between pairs[4] while keeping the rest of the algorithm identical. The performance gap between our method and the baseline shows that the relation learning method, apart from structured inference formulation, plays an important role in boosting the performance. See appendix for comparison of different structure inference algorithms.

5.2.8 Co-Localization

We evaluate on the co-localization problem to illustrate the benefits of the methods discussed in the chapter on a real world and large scale dataset. In this task, we train the algorithm on a split of COCO 2017 (Lin et al., 2014b) dataset with 63 seen classes and evaluate on the remaining 17 unseen classes. The resulting dataset contains $111,085$ and $8,245$

[4]We also use negative of Euclidean distance measure for the relation but it shows inferior performance.

Table 5.2: CorLoc(%) on COCO and ImageNet with 8 positive and 8 negative images.

Method	COCO	ImageNet
MI-SVM (Hoffman et al., 2015)	60.74 ± 1.07	49.44 ± 1.10
ATNMIL (Ilse, Tomczak, and Welling, 2018)	60.00 ± 1.07	49.35 ± 1.10
Ours+TRWS	65.04 ± 1.05	54.20 ± 1.09
Ours+Greedy	$\mathbf{65.34 \pm 1.04}$	$\mathbf{55.18 \pm 1.09}$
Ours+AStar	64.99 ± 1.05	54.23 ± 1.09
Unary Only	59.24 ± 1.08	50.29 ± 1.10
Ours+TRWS Pairwise Only	64.53 ± 1.05	52.95 ± 1.09
Ours+AStar Pairwise Only	64.54 ± 1.05	52.89 ± 1.09
Ours+Greedy Pairwise Only	64.65 ± 1.05	53.00 ± 1.10

images in the training and test set respectively. To evaluate the performance of the trained algorithm on a larger set of unseen classes we also test on validation set of ILSVRC2013 detection (Russakovsky et al., 2015). This dataset has originally 200 classes but only 148 classes do not have overlap with the classes that were used for training. The final dataset, after removing coco seen classes, contains $12,544$ images from 148 unseen classes. The dataset creation method is explained in the supplementary material in more detail.

For the CNN feature extractor module, we pre-train a Faster-RCNN detector (Girshick, 2015) with ResNet-50 (He et al., 2016) backbone on the COCO training dataset which has only seen classes. For each image, region proposals with the highest objectness scores are kept. The output of the second stage feature extractor is used in all methods.

For this task, each bag is constructed by extracting top $B = 300$ region proposals from one image and a selection O represents one bounding box from each image. To select images of each problem, we first randomly select one class as the target. Then, N images which have at least one object from the target class are sampled as positive bags. The negative bag is composed of images which do not contain the target class. The success rate metric used in few-shot common object detection is used to evaluate the performance of different algorithms. A region proposal is considered successful if it has IoU overlap greater than 0.5 with the ground-truth target bounding box. Note that for the co-localization task, this metric is equivalent to class agnostic CorLoc (Deselaers, Alexe, and Ferrari, 2010) measure which

is widely used for localization problem evaluation (Bilen, Pedersoli, and Tuytelaars, 2015; Cinbis, Verbeek, and Schmid, 2017; Shi, Caesar, and Ferrari, 2017; Uijlings, Popov, and Ferrari, 2018).

Table 5.2 illustrates the quantitative results on COCO and ImageNet datasets with 8 positive and 8 negative images[5]. In addition to TRWS (Kolmogorov, 2006), we use Astar (Bergtholdt et al., 2010) and greedy matching (Shaban et al., 2019a) for the inference. Our method works considerably better than other strong MIL baselines. Qualitative results of our method compared with other MIL based approaches are illustrated in Figure 5.3. Our method selects the correct object even when the target object is not salient. More qualitative results are presented in the supplementary material.

To see the effect of unary and pairwise potentials separately, we provide results for two new variants for structured inference based methods: (i) Unary Only: where the common object proposal in each bag is selected using only the information in negative bags without seeing the elements in other bags, and (ii) Pairwise Only: where the negative bag information is ignored in each problem. The results show that the pairwise potentials contribute more to the final results. This is not surprising since negative images only help when they contain an object which is also appearing in positive images which, given the number of classes we are sampling from, has a low chance. Interestingly, by using the learned unary potentials alone we could get comparable results to the MIL baselines. The results in Table 5.2 show that different inference algorithms have very similar performance. However, as it is shown in Figure 5.4, the greedy matching algorithm of (Shaban et al., 2019a) is much faster. Note that the greedy optimization algorithm requires to compute only 15% out of all pairwise potentials in average.

[5]We skip the results for sbMIL and mi-SVM as they showed similar or inferior results to MI-SVM.

Figure 5.3: *Qualitative results on COCO dataset. Each row shows positive bags of a sampled collection. Negative bags are not shown. Note that the first image in the first two rows are identical but the target object is different. Last row shows a failure case of our algorithm. While cup is the target object, our method finds plant in the second image. This might be due to the fact that pot (which has visual similarities to cup) and plant are labelled as one class in the training dataset. Note that "dog", "cake" and "cup" are samples from unseen classes. Selected regions are tagged with method names. Ground-truth target bounding box is shown in green with tag "GT".*

5.2.9 One-Shot Image Recognition

We evaluate the quality of the learned pairwise relations $r(e, e')$ by using them for the task of one-shot image recognition (Vinyals et al., 2016) on *mini*ImageNet and compare it to the other state-of-the-art methods. In each episode of a one-shot 5-way problem, 5 classes are randomly chosen from the set of possible classes and one image is sampled from each class. This mini-training set is used to predict the label of a new query image which is sampled from one of the 5 classes. The performance is the accuracy of the method to predict the correct label averaged over many sampled episodes. All of these models are trained with a variant of deep residual networks (He et al., 2016; Zagoruyko and Komodakis, 2016). Note that unlike other methods, the model in (Qiao et al., 2018) is trained on validation+training meta-sets.

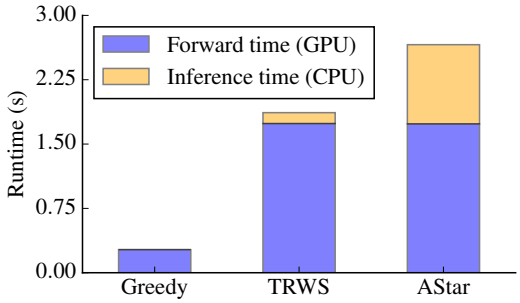


Figure 5.4: *Forward and inference time (in sec.) on COCO.*

Table 5.3: 5-way, 1-shot, classification accuracy with 95% confidence interval on *mini*ImageNet test set.

Method	Accuracy (%)
adaResNet (Munkhdalai et al., 2018)	56.88 ± 0.62
SNAIL (Mishra et al., 2018)	55.71 ± 0.99
Gidaris et al. (Gidaris and Komodakis, 2018)	55.45 ± 0.89
TADAM (Oreshkin, Lacoste, and Rodriguez, 2018)	58.50 ± 0.30
Qiao et al. (Qiao et al., 2018)	$\mathbf{59.60 \pm 0.41}$
Ours-ReLU	56.43 ± 0.79
Ours-Gated	57.80 ± 0.77

At each episode, we use the learned relation function to score the similarity between the query image and all the images in the mini training set. The predicted label for the query image is simply the label of the image in mini training set which has the highest relation value to the query image. We compute the accuracy of the predictions of our pairwise potentials on test classes of *mini*ImageNet and compare it with current state-of-the-art few-shot methods in Table 5.3. We also provide comparison of gated activation function and a simplified ReLU activation in our architecture. Although our method is not trained directly for the task of one-shot learning, it achieves competitive results to the previous methods which are specifically trained for the task. Also, the results show the advantage of using gated activation over ReLU.

5.3 Extension to Large Scale Weakly Supervised Object Localization

Thus far, we only consider the problem of co-localization when there are only few positive bags in each problem. In this section, we discuss a plausible method to extend the proposed method to large scale datasets. In these situations, finding an approximate labeling quickly becomes impractical, since the number of terms in Eq 5.2 increases quadratically with the numbers of positive bags in $\mathcal{W}$ due to dense pairwise connectivity.

Due to this limitation, we employ an older well-known iterated conditional modes (ICM) algorithm for optimization (Besag, 1986). In each step, ICM only updates the labeling of a single bag while all the other labeling are fixed. It is known that ICM generates monotonically non-increasing objective values and is computationally efficient. However, since ICM performs coordinate descent type updates and the problem in Eq 5.2 is neither convex nor differentiable, ICM is prone to get stuck at a local minimum and its solution significantly depends on the quality of the initial labeling.

The experiments in Section 5.2.8 has shown that using the pairwise and unary functions learned on the source dataset, the co-localization method performs reasonably well by only looking at few bags. Motivated by this, we divide the full size problem into a set of disjoint

mini-problems, solve each mini-problem efficiently using a state-of-the-art TRWS inference algorithm, and use these results to initialize the ICM algorithm.

The initialization algorithm samples a mini-problem $\mathcal{X} \in \mathcal{V}$ and optimizes the co-localization problem over $\mathcal{X}$. This process is repeated until all the bags in the dataset are covered. The complete large co-localization step is illustrated in Algorithm 1.

Next, we analysis the time complexity of the re-localization step. We practically observed that computing the pairwise similarity scores is the computation bottleneck, thus we analyze the time complexities in terms of the number of pairwise similarity scores each algorithm computes. Let $N = |\mathcal{V}|$ denotes the number of positive bags, and $B = \max_{\mathcal{B}\in\mathcal{V}} |\mathcal{B}|$ be the maximum bag size. To solve the exact optimization in Eq 5.2, we need to compute $\mathcal{O}(B^2N^2)$ pairwise elements. On the other hand, each iteration of ICM only computes $\mathcal{O}(BN)$ elements and we have to compute the total of $\mathcal{O}(NKB^2)$ pairwise similarity scores for the initialization where K is the size of the mini-problem. Thus, ICM algorithm would be asymptotically more efficient than the exact optimization in terms of total number of pairwise similarity scores it computes, if it is run for $\Omega(NB)$ iterations or $E = \Omega(B)$ epochs. We practically observe that by initializing ICM with the result of the proposed initialization scheme it convergences in few epochs.

Algorithm 1: large scale co-localization.

Input: Set of positive bags $\mathcal{V}$, mini-problem size K, #epochs E
Output: Optimal selection O^*
$T \leftarrow round(\frac{|\mathcal{V}|}{K})$, randomly initialize O
for $t \leftarrow 1$ ***to*** T **do**
  `// Sample next mini-problem`
  $\mathcal{X} \sim \mathcal{V}$
  `// Solve mini-problem with TRWS Kolmogorov, 2006`
  $\bar{O}^* \leftarrow \arg\min_{\bar{O}} \sum_{\substack{e_i,e_j\in\bar{O} \\ i>j}} \psi_\theta^{\mathrm{P}}(e_i, e_j) + \eta \sum_{e_i\in O} \psi_\beta^{\mathrm{U}}(e_i \mid \bar{v})$
  Update corresponding block of O with $\bar{O}^*$
`// Finetune for` E `epochs`
$O^* \leftarrow \mathtt{ICM}(O, E)$
return O^*

5.3.1 Experiments

We evaluate our algorithm on the large scale object localization to illustrate the benefits of utilizing the proposed method for initializing ICM algorithm. We use the similar data and networks as the few-shot co-localization experiment in Section 5.2.8. To create large scale problems, for each unseen class, we choose all the images in the COCO test dataset that have at least one instance in that class. This results in a total of 17 different problems. For simplicity, we does not use any negative bag in these experiments i.e. $\eta = 0$ in Algorithm 1. We run the proposed algorithm on each of the problem and report the mean CorLoc over all 17 problems.

For this experiment, we initialize the labeling of the images of each problem using the following initialization strategies:

- Random: randomly select a proposal from each bag.
- Objectness: select the proposal with the highest unary score from each bag.
- Proposed initialization method: Proposed initialization method discussed in Algorithm 1. We conduct the experiment with different mini-problem sizes $K \in \{2, 4, 8, 64\}$. We use the state-of-the-art TRWS (Kolmogorov, 2006) algorithm for inference in each mini-problem.

Finally, we perform ICM with each of the initialization methods. Fig 5.5 shows the CorLoc and Energy vs. time plots as well as the computation time for different initialization methods. The results show that $K = 64$ exhibits the best initialization performance. However, ICM converges to similar energy for $4 \leq K \leq 64$. In the extreme case with mini-problem of size $K = 2$, ICM converges to a worse local minimum in terms of CorLoc and energy value. Surprisingly, objectness and $K = 2$ initialization does not work better than random initialization. We also tried initializing ICM with the proposal that covers the complete image as it is the initialization scheme that is commonly used in MIL alternating

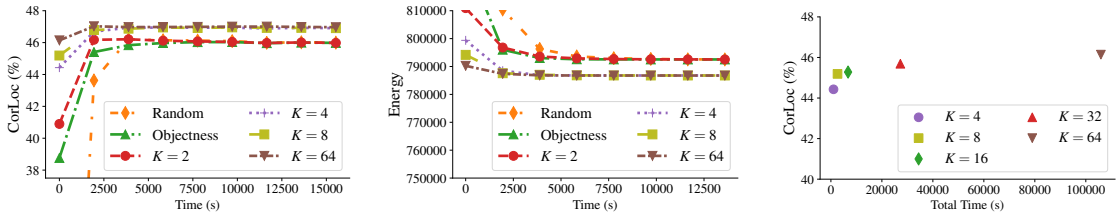


Figure 5.5: **Left:** ICM CorLoc(%) vs. time for different initialization methods. See initialization schemes for definition of each initialization method. Markers indicate start of a new epoch. ICM inference convergences in 2 epochs and demonstrates its best performance when is initialized with the proposed initialization method. **Middle:** Energy vs. time for different initialization methods. The energies in the plot are computed by summing over energies of all classes. **Right:** Runtime vs. CorLoc(%) comparison of the proposed initialization scheme with various mini-problem sizes.

optimization algorithms (Cinbis, Verbeek, and Schmid, 2017; Uijlings, Popov, and Ferrari, 2018). Unfortunately, this method produces significantly worse results than the other methods and hence we omit it in this experiment.

These results highlight the importance of initialization in ICM inference. Fortunately, ICM can effectively enhance the result of small size mini-problems in just few epochs. Note that increasing K beyond 64 might still provide a better initialization to ICM and increase the results further. Thus, one should increase the mini-problem size as far as time and computational resources allow.

5.4 Few-Shot Weakly Supervised Object Detection

Few-shot semantic segmentation problem discussed in Chapter 3 is an instance of supervised few-shot learning since the support set is fully annotated. In this section, we introduce few-shot weakly supervised detection problem where only image level labels of images in the support set are available. Given the weakly supervised support set, the task is to detect the target object in an query image. When the image annotations are not available, one could use the proposed co-localization method to find the common objects in the support set. Once the support set is annotated, any off-the-shelf supervised few-shot detection algorithm can be used to detect the object in the query image.

5.4.1 Problem Setup

Readers might need to review Chapter 3 to get familiar with some of the terms we use here to describe the few-shot detection problem. Similar to co-localization problem, we represent each image by a bag of image proposals. We define a few-shot weakly supervised problem by collection $\mathcal{U} = (\mathcal{S}, \mathcal{L}, \mathcal{Q})$ where $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_N)$ is list of bags in the support set, $\mathcal{L} = (\mathcal{L}_1, \ldots, \mathcal{L}_N)$ is their corresponding bag labels, and $\mathcal{Q}$ is a query bag. We use notation $(\mathcal{S}, \mathcal{L}, \mathcal{Q}) \sim \mathcal{D}$ to indicate that a random weakly supervised detection problem is sampled from dataset $\mathcal{D}$. Unlike the supervised counterpart, we do not have access to the label of images in support bags. Instead, set $\mathcal{L}_i \subseteq \mathscr{C}$ only shows foreground classes that are present in bag $\mathcal{S}_i$ where $\mathscr{C}$ is set of all foreground classes in $\mathcal{D}$.

The goal is to learn a model that, when given $\mathcal{U}$, predicts label of all image proposals $e \in \mathcal{Q}$. An image in a query bag should be classified as a foreground object with its correct label if an instance of that object is present in the support set and background otherwise. Let $c_e \in \mathscr{C}$ be the true label of image e and $\mathcal{F} = \{c \mid c \in \mathcal{L}_i \quad \forall i \in \{1, \ldots, N\}\}$ denotes the set of all foreground classes present in $\mathcal{S}$, we define label of e with respect to $\mathcal{F}$ as

$$c_e^{\mathcal{F}} = \begin{cases} c_e & c_e \in \mathcal{F} \\ c_\varnothing & \text{otherwise,} \end{cases} \tag{5.5}$$

where $c_\varnothing$ is background class.

We solve the problem of few-shot weakly supervised object detection in a two stage process. In the first stage, we use the proposed co-localization method to annotate the support set images. Once the support set is annotated, we can use any off-the-shelf few-shot object detection algorithm to detect foreground objects in the query images. We describe these steps in the following sections.

5.4.2 Support Set Annotation

We use the co-localization method proposed in Section 5.2 to annotate images in the support set. For each class $c \in \mathcal{F}$, we define co-localization problem $\mathcal{W}_c = (\mathcal{S}_c, \bar{\mathcal{S}}_c)$ where $\mathcal{S}_c = \{\mathcal{S}_i \mid \mathcal{S}_i \in \mathcal{S} \text{ if } c \in \mathcal{L}_i\}$ is a positive (consistent) collection with respect to class c and $\bar{\mathcal{S}}_c = concat(\mathcal{S} \setminus \mathcal{S}_c)$ is a negative bag formed by concatenating all the proposals in the negative bags $\mathcal{S} \setminus \mathcal{S}_c$ into a single bag $\bar{\mathcal{S}}_c$.

Let O^c be a selection that minimizes the energy function in Eq 5.2 that is defined with respect to the collection $\mathcal{W}_c$. Using only the top selection O^c may not be optimal for two reasons. First, as shown in the experiment of Section 5.2.8, the top selection has a high error rate and there is a high chance that the correct image does not get selected by the top selection due to the ambiguities in finding the correct common object across few bags. Second, when there are more than one instance of an object in a support bag, top selection one capture one of them in the best case. To allow more flexibility to the algorithm, we extract several qualitatively distinct selections proposals from each co-localization problem. For this purpose, we utilize M-best Mode algorithm in (Batra et al., 2012) to extract top M qualitatively distinct selections that minimizes the the energy function $E(O \mid \bar{\mathcal{S}}_c)$. M-best Mode algorithm is a greedy method that iteratively finds new distinctive selections. In each iteration, the algorithm finds a new selection that is distinct from the set of current selection and minimizes the energy function $E(O \mid \bar{\mathcal{S}}_c)$. Let $\mathcal{O}^c = \{O^c_1, \ldots, O^c_j\}$ be the set of current selections at jth iterations, next best selection is found by solving

$$\begin{aligned} O^c_{j+1} = &\arg\min_{O} E(O \mid \bar{\mathcal{S}}_c), \\ &\text{s.t.} \quad \Delta(O, O^c_i) \geq \beta \quad \forall O^c_i \in \mathcal{O}^c \end{aligned} \tag{5.6}$$

where O is a selection over positive collection $\mathcal{S}_c$, $\Delta(.,.)$ is a dissimilarity function measuring the distance of two selections, and scalar β controls how far a new solution has to be from the current ones. We measure the dissimilarity of two selection by counting the number of

locations at which the their corresponding images are different i.e.

$$\Delta(O, O') = \sum_i \delta(e_i \neq e'_i), \tag{5.7}$$

where e_i and e'_i are the ith selected images in O and O' respectively.

Following (Batra et al., 2012), we optimize a specific form of the continuous relaxation of the optimization in Eq 5.6, formed by dualizing the dissimilarity constraint

$$O^c_{j+1} = \arg\min_O E(O \mid \bar{\mathcal{S}}_c) - \lambda \sum_{O^c_i \in \mathcal{O}^c} \Delta(O, O^c_i), \tag{5.8}$$

where O^c_{j+1} minimizes a linear combination of the energy and similarity (negative dissimilarity) to the current selections. As it is shown in (Batra et al., 2012), for the dissimilarity function in Eq 5.7, the optimization becomes the same as the original energy minimization with modified unary potentials. Formally, optimization in Eq 5.8 is equivalently written as

$$E(O \mid \bar{v}) = \sum_{\substack{e_i, e_j \in O \\ i>j}} \psi^{\mathrm{P}}_\theta(e_i, e_j) + \eta \sum_{e_i \in O} \big(\psi^{\mathrm{U}}_\beta(e_i \mid \bar{v}) + \lambda \sum_{O^c \in \mathcal{O}^c} \delta(e_i = e'_i)\big), \tag{5.9}$$

where e'_i is the ith element in O^c. Augmenting the unary potentials in this way biases away next selection from the current selections. We can find M-Best Modes simply by using the same structure inference algorithms used for the co-localization problem by just modifying the unary potentials before feeding them into any of these algorithms.

5.4.3 Few-shot Object Detection

Having a set of selection proposals $\mathcal{O}^c$ for each class $c \in \mathcal{F}$ present in the support bags, we propose a simple algorithm for classifying images in query bag $\mathcal{Q}$. Although we propose a specific architecture for the task, any other supervised few-shot object detection methods can be adapted for this purpose as the techniques we use here are generally applicable to any probabilistic classifiers.

Let $e \in \mathcal{Q}$ be an image in query bag $\mathcal{Q}$, given all selection proposals $\{\mathcal{O}^c\}_{c\in\mathcal{F}}$, our goal is to classify e to one of the foreground classes in $\mathcal{F}$ or background class $c_\varnothing$. Formally, we learn a probabilistic classifier $Pr(Y|e, \{\mathcal{O}^c\}_{c\in\mathcal{F}})$ where label Y can take any value from the set $\mathcal{F} \cup \{c_\varnothing\}$.

We do this by learning a scoring function $g : \mathcal{I} \times \mathcal{I}^N \rightarrow \mathbb{R}$ that computes similarity of an image $e \in \mathcal{I}$ and a selection $O \in \mathcal{I}^N$ where N is length of selection O. Let $\mathcal{P}^c(e) = \{g(e, O_1^c), \ldots, g(e, O_M^c)\}$ be set that contains similarity of image e to every selection proposal in $\mathcal{O}^c$. Each selection is a candidate for a correct localization of objects in class c. Among these candidates, we pick the one that is most similar to the query image e. Thus, the class score of image e for foreground class c is computed as $p^c(e) = \max(\mathcal{P}^c(e))$. Additionally, we need to assign a background class score $p^{c\varnothing}(e)$ to image e. We do this by simply assigning background score zero to all the query images i.e. $p^{c\varnothing}(e) = 0$. By using zero threshold for the background class, function g is learned to predict a positive value if the image is most likely from a foreground object captured in selection O and negative otherwise. Finally, we compute the probabilistic classification output using softmax operation

$$Pr(Y = c|e, \{\mathcal{O}^c\}_{c\in\mathcal{F}}) = \frac{\exp p^c(e)}{\sum_{c\in\mathcal{F}\cup c_\varnothing} \exp p^c(e)}. \tag{5.10}$$

Function g Architecture The scoring function g predicts the similarity of input image e and selection O in few stages. Let $f \in \mathbb{R}^d$ and $f_i \in \mathbb{R}^d$ be the features extracted from images e and $e_i \in O$ respectively. First, we use an embedding function $\Theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ to map f and $f_i \in O$ to a d-dimensional space. For the experiments, function Θ is built by a two layer fully connected neural network with ReLU activation functions. Next, we use average operation to find a d-dimensional representation of selection O

$$\bar{O} = \frac{1}{N}\sum_{i=1}^{N} \Theta(f_i). \tag{5.11}$$

Finally, we use relation network proposed in Section 5.2.3 to estimate the similarity of $\Theta(e)$

and $\bar{O}$.

Training We train the model end-to-end. First, we sample randomly sample a weakly supervised object detection problem $\mathcal{U} = (\mathcal{S}, \mathcal{L}, \mathcal{Q})$ from the training dataset i.e. $\mathcal{U} \sim \mathcal{D}_{\text{train}}$. Then, we find qualitatively diverse selections for each foreground class using the algorithm in Section 5.4.2. Finally, we use these selection proposals to compute the probabilistic class prediction $Pr(Y|e, \{\mathcal{O}^c\}_{c\in\mathcal{F}})$ for every image $e \in \mathcal{Q}$. We define the loss function as

$$\mathcal{L} = \mathbb{E}_{\mathcal{U}\sim\mathcal{D}_{\text{train}}}\big(\frac{1}{Z}\sum_{e\in\mathcal{Q}} \ell(Pr(Y|e, \{\mathcal{O}^c\}_{c\in\mathcal{F}}), c_e^{\mathcal{F}})\big), \tag{5.12}$$

where $c_e^{\mathcal{F}}$ is the ground-truth label for image e with respect to $\mathcal{F}$, Z is the total number of images in the query bag $\mathcal{Q}$, and ℓ is the cross-entropy loss function. Note that the co-localization model that is used to find selection proposals is trained first and remains fixed during the training of the few-shot detector model.

5.4.4 Experiments

We evaluate on the few-shot weakly supervised object detection problem to illustrate the performance of the proposed algorithm. We use the similar splits of COCO 2017 dataset and CNN feature extractor module that were used in the co-localization experiments. We also use the same model trained for the co-localization task to perform support set annotation.

To sample an instance of few-shot weakly supervised object detection, we first randomly select n classes. For of these classes, we sample k images which have at least one object from the target class. This create a support set $\mathcal{S}$ with nk images where there are at least k images for each class. We sample query bag by randomly selecting an image that have at least one object in any of the n selected classes. Finally, we construct bags by extracting top $B = 100$ region proposals from each image. For all the experiments, we report the Mean Average Precision (MAP) which is a popular metric for object detection task.

In addition to the proposed weakly supervised method, we report the results for the

Table 5.4: The performance of the proposed weakly supervised detection for $k = 1$ and $n = 5$ for different number of selection proposals.

Method	M-Best	MAP@0.5 (%)
Supervised	1	10.45
Weakly Supervised	1	8.84
Weakly Supervised	5	9.26
Weakly Supervised	10	**9.31**
Weakly Supervised	15	9.30
Weakly Supervised	20	9.30

Table 5.5: The performance of the proposed weakly supervised detection for $k = 5$ and $n = 5$ for different number of selection proposals.

Method	M-Best	MAP@0.5 (%)
Supervised	1	16.73
Whole Image	1	9.21
Unary Only	1	13.38
Weakly Supervised	1	15.69
Weakly Supervised	2	16.45
Weakly Supervised	3	16.51
Weakly Supervised	4	16.55
Weakly Supervised	5	16.55
Weakly Supervised	6	16.53
Weakly Supervised	7	16.54

supervised few-shot detection method where the ground-truth annotation is directly fed into the few-shot detection model.

Table 5.4 and 5.5 illustrate the quantitative results on COCO dataset for $n = 5$ and $k = \{1, 5\}$ with different number of selection proposals. Interestingly, in the few-shot setting, the performance gap between weakly supervised approach and the supervised counterpart is small compared to the gap in large-scale weakly supervised detection. The results also show that choosing increasing the number of selections helps in closing the gap between supervised and weakly supervised detection methods.

5.5 Conclusion

We introduce a method for learning to find images of a common object category across few bags of images which is constructed by learning unary and pairwise terms in an structured output prediction framework. Moreover, we propose an inference algorithm that uses the structure of the problem to solve the task at hand without requiring computation of all pairwise terms. Our experiments on two challenging tasks in the low data regime illustrate the advantage of our knowledge transfer method to several MIL weakly supervised algorithms. In addition, our inference algorithm performs comparable to the well-known structured inference algorithms for this task while being faster.

Appendices

APPENDIX A

TRUNCATED BACK-PROPAGATION FOR BILEVEL OPTIMIZATION

A.1 Proof of Proposition 2.3.1

Proposition 2.3.1. *Assume g is β-smooth, twice differentiable, and locally α-strongly convex in w around $\{w_{T-K-1}, \dots, w_T\}$. Let $\Xi_{t+1}(w_t, \lambda) = w_t - \gamma \nabla_w g(w_t, \lambda)$. For $\gamma \leq \frac{1}{\beta}$, it holds*

$$\|h_{T-K} - \mathrm{d}_\lambda f\| \leq 2^{T-K+1}(1-\gamma\alpha)^K \|\nabla_{\hat{w}^*} f\| M_B \tag{2.8}$$

where $M_B = \max_{t\in\{0,\dots,T-K\}} \|B_t\|$. In particular, if g is globally α-strongly convex, then

$$\|h_{T-K} - \mathrm{d}_\lambda f\| \leq \tfrac{(1-\gamma\alpha)^K}{\gamma\alpha} \|\nabla_{\hat{w}^*} f\| M_B. \tag{2.9}$$

Proof. Let $\mathrm{d}_\lambda f - h_{T-K} = e_K$. By definition of h_{T-K},

$$e_K = \left(\sum_{t=0}^{T-K} B_t A_{t+1} \cdots A_{T-K} \right) A_{T-K+1} \cdots A_T \nabla_{\hat{w}^*} f$$

Therefore, when g is locally α-strongly convex with respect to w in the neighborhood of $\{w_{T-K-1}, \dots, w_T\}$,

$$\begin{aligned} \|e_K\| &\leq \| \sum_{t=0}^{T-K} B_t A_{t+1} \cdots A_{T-K} \| \| A_{T-K+1} \cdots A_T \nabla_{\hat{w}^*} f \| \\ &\leq (1-\gamma\alpha)^K \|\nabla_{\hat{w}^*} f\| \| \sum_{t=0}^{T-K} B_t A_{t+1} \cdots A_{T-K} \| \end{aligned}$$

Suppose g is β-smooth but nonconvex. In the worst case, if the smallest eigenvalue of $\nabla_{w,w} g(w_{t-1}, \lambda)$ is $-\beta$, then $\|A_t\| = 1 + \gamma\beta \leq 2$ for $t = 0, \dots, T-K$. This gives the

bound in (2.8). However, if g is globally strongly convex, then

$$\|e_K\| \leq \|\nabla_{\hat{w}^*} f\|(1-\gamma\alpha)^K \max_{t\in\{0,\ldots,T-K\}} \|B_t\| \sum_{t=0}^{T-K} (1-\gamma\alpha)^t$$

The bound (2.9) uses the fact that $\sum_{t=0}^{T-K}(1-\gamma\alpha)^t \leq \sum_{t=0}^{\infty}(1-\gamma\alpha)^t = \frac{1}{\gamma\alpha}$ ■

A.2 Proof of Lemma 2.3.2

Lemma 2.3.2. *Let g be globally strongly convex and $\nabla_\lambda f = 0$. Assume g is second-order continuously differentiable and B_t has full column rank for all t. Let $\Xi_{t+1}(w_t, \lambda) = w_t - \gamma\nabla_w g(w_t, \lambda)$. For all $K \geq 1$, with T large enough and γ small enough, there exists $c > 0$, s.t. $h_{T-K}^\top \mathrm{d}_\lambda f \geq c\|\nabla_{\hat{w}^*} f\|^2$. This implies h_{T-K} is a sufficient descent direction, i.e. $h_{T-K}^\top \mathrm{d}_\lambda f \geq \Omega(\|\mathrm{d}_\lambda f\|^2)$.*

Proof. To illustrate the idea, here we prove the case where $K = 1$. For $K > 1$, similar steps can be applied. To prove the statement, we first expand the inner product by definition

$$h_{T-1}^\top \mathrm{d}_\lambda f = \|h_{T-1}\|^2 + (B_T\nabla_{\hat{w}^*} f)^\top \left(\sum_{t=0}^{T-1} B_t A_{t+1}\cdots A_{T-1}\right) A_T \nabla_{\hat{w}^*} f$$

where we recall $h_{T-1} = B_T\nabla_{\hat{w}^*} f$ as $\nabla_\lambda f = 0$ by assumption.

Next we show a technical lemma, which provides a critical tool to bound the second term above; its proof is given in the next section.

Lemma A.2.1. *Let g be α-strongly convex and β-smooth. Assume B_t and A_t are Lipschitz continuous in w, and assume B_T has full column rank. For $\gamma \leq \frac{1}{\beta}$,*

$$\begin{aligned}
&(B_T\nabla_{\hat{w}^*} f)^\top B_t A_{t+1}\cdots A_T \nabla_{\hat{w}^*} f \\
&\geq (1-\gamma\alpha)^{T-t}\|B_T\nabla_{\hat{w}^*} f\|^2 - \|\nabla_{\hat{w}^*} f\|^2 O\left(\frac{e^{-\alpha\gamma(T-1)}}{1-e^{-\alpha\gamma}} + (\gamma(\beta-\alpha))^{T-t}\right)
\end{aligned}$$

By Lemma A.2.1, we can then write

$$h_{T-1}^{\top}\mathrm{d}_{\lambda}f \geq \|B_T\nabla_{\hat{w}^*}f\|^2\left(1+\sum_{t=0}^{T-1}(1-\gamma\alpha)^{T-t}\right)-\|\nabla_{\hat{w}^*}f\|^2 O\left(\sum_{t=0}^{T-1}\frac{e^{-\alpha\gamma(T-1)}}{1-e^{-\alpha\gamma}}+(\gamma(\beta-\alpha))^{T-t}\right)$$

Because

$$\sum_{t=0}^{T-1}(\gamma(\beta-\alpha))^{T-t}=\sum_{k=1}^{T}(\gamma(\beta-\alpha))^{k}\leq\frac{\gamma(\beta-\alpha)}{1-\gamma(\beta-\alpha)} \qquad (\because \gamma\leq\beta)$$

and $B_T^{\top}B_T$ is non-singular by assumption,

$$\begin{aligned} h_{T-1}^{\top}\mathrm{d}_{\lambda}f &\geq \|\nabla_{\hat{w}^*}f\|^2\Omega(1)-\|\nabla_{\hat{w}^*}f\|^2 O\left(\frac{Te^{-\alpha\gamma(T-1)}}{1-e^{-\alpha\gamma}}+\frac{\gamma(\beta-\alpha)}{1-\gamma(\beta-\alpha)}\right) \\ &\geq C\|\nabla_{\hat{w}^*}f\|^2 \end{aligned}$$

for some $c>0$, when T is large enough and γ is small enough. The implication holds because $\|\mathrm{d}_{\lambda}f\|\leq O(\|\nabla_{\hat{w}^*}f\|)$. ■

A.2.1 Proof of Lemma A.2.1

Proof. Let C_A and C_B be the Lipschitz constant of A_t and B_t. First, we see that the inner product can be lower bounded by the following terms

$$\left(B_T\nabla_{\hat{w}^*}f\right)^{\top}B_tA_{t+1}\cdots A_T\nabla_{\hat{w}^*}f\geq(1-\gamma\alpha)^{T-t}\|B_T\nabla_{\hat{w}^*}f\|^2-\Delta_1-\Delta_2-\Delta_3$$

where

$$\Delta_1=C_B\|B_T\nabla_{\hat{w}^*}f\|\|\nabla_{\hat{w}^*}f\|\|w_{T-1}-w_{t-1}\|\|A_{t+1}\cdots A_T\|$$

$$\Delta_2=C_A\|B_T^{\top}B_T\nabla_{\hat{w}^*}f\|\|\nabla_{\hat{w}^*}f\|\sum_{k=t+1}^{T-1}\|w_{T-1}-w_{k-1}\|\|A_{t+1}\cdots A_{k-1}\|\|A_T\|^{T-k}$$

$$\Delta_3 = \|\nabla_{\hat{w}^*} f\| \|B_T^\top B_T \nabla_{\hat{w}^*} f\| \|A_k - (1-\gamma\alpha) I\|^{T-k}$$

The above lower bounds can be shown by the following inequalities:

$$\begin{aligned}
&\left(B_T \nabla_{\hat{w}^*} f\right)^\top B_t A_{t+1} \cdots A_T \nabla_{\hat{w}^*} f \\
&\geq \nabla_{\hat{w}^*} f^\top \left(B_T^\top B_T\right) A_{t+1} \cdots A_T \nabla_{\hat{w}^*} f - C_B \|B_T \nabla_{\hat{w}^*} f\| \|w_{T-1} - w_{t-1}\| \|A_{t+1} \cdots A_T \nabla_{\hat{w}^*} f\|
\end{aligned}$$

$$\begin{aligned}
&\nabla_{\hat{w}^*} f^\top \left(B_T^\top B_T\right) A_{t+1} \cdots A_T \nabla_{\hat{w}^*} f \\
&\geq \nabla_{\hat{w}^*} f^\top \left(B_T^\top B_T\right) A_{t+1} \cdots A_{T-2} A_T^2 \nabla_{\hat{w}^*} f \\
&- C_A \|w_{T-1} - w_{T-2}\| \|A_{t+1} \cdots A_{T-2}\| \|A_T\| \|B_T^\top B_T \nabla_{\hat{w}^*} f\| \|\nabla_{\hat{w}^*} f\| \\
&\geq \nabla_{\hat{w}^*} f^\top B_T^\top B_t A_T^{T-t} \nabla_{\hat{w}^*} f \\
&- C_A \|B_T^\top B_T \nabla_{\hat{w}^*} f\| \|\nabla_{\hat{w}^*} f\| \sum_{k=t+1}^{T-1} \|w_{T-1} - w_{k-1}\| \|A_{t+1} \cdots A_{k-1}\| \|A_T\|^{T-k}
\end{aligned}$$

$$\begin{aligned}
\nabla_{\hat{w}^*} f^\top B_T^\top B_T A_T^{T-t} \nabla_{\hat{w}^*} f \geq\ & (1-\gamma\alpha)^{T-t} \nabla_{\hat{w}^*} f^\top B_T^\top B_T \nabla_{\hat{w}^*} f \\
& - \|\nabla_{\hat{w}^*} f\| \|B_T^\top B_T \nabla_{\hat{w}^*} f\| \|A_T - (1-\gamma\alpha) I\|^{T-t}
\end{aligned}$$

Next we upper bound the error terms: Δ_1, Δ_2, and Δ_3. We will use the fact that gradient descent converges linearly when optimizing a strongly convex and smooth function Hazan, 2016.

Lemma A.2.2. *Let w_0 be the initial condition. Running gradient descent to optimize an α-strongly convex and β-smooth function g, with step size $0 < \gamma \leq \frac{1}{\beta}$, generates a sequence $\{w_t\}$ satisfying*

$$\|w_t - w^*\| \leq D e^{-\alpha\gamma t} \tag{A.1}$$

where $D = \|w_0 - w^*\|$ *and* $w^* = \arg\min g(w)$.

Lemma A.2.2 implies for $T \geq t$, $\|w_T - w_t\| \leq 2De^{-\alpha\gamma t}$.

Now we proceed to bound the errors Δ_1, Δ_2, and Δ_3.

Bound on Δ_1 Because

$$\begin{aligned}\|w_{T-1} - w_{t-1}\|\|A_{t+1}\cdots A_T\| &\leq 2De^{-\alpha\gamma(t-1)}(1-\gamma\alpha)^{T-t} \\ &\leq 2De^{-\alpha\gamma(t-1)}e^{-\gamma\alpha(T-t)} \\ &= 2De^{-\alpha\gamma(T-1)}\end{aligned}$$

we can upper bound Δ_1 by

$$\begin{aligned}\Delta_1 &= C_B\|B_T\nabla_{\hat{w}^*}f\|\|\nabla_{\hat{w}^*}f\|\|w_{T-1} - w_{t-1}\|\|A_{t+1}\cdots A_T\| \\ &\leq \|B_T\nabla_{\hat{w}^*}f\|\|\nabla_{\hat{w}^*}f\| \times 2C_BDe^{-\alpha\gamma(T-1)}\end{aligned}$$

Bound on Δ_2 Because

$$\begin{aligned}\sum_{k=t+1}^{T-1}\|w_{T-1} - w_{k-1}\|\|A_{t+1}\cdots A_{k-1}\|\|A_T\|^{T-k} &\leq \sum_{k=t+1}^{T-1} 2De^{-\alpha\gamma(k-1)}(1-\alpha\gamma)^{k-1-t+T-k} \\ &\leq 2D(1-\alpha\gamma)^{T-t-1}\sum_{k=t+1}^{T-1}e^{-\alpha\gamma(k-1)} \\ &\leq 2D(1-\alpha\gamma)^{T-t-1}e^{-\alpha\gamma t}\sum_{k=t+1}^{T-1}e^{-\alpha\gamma(k-t-1)} \\ &\leq 2De^{-\alpha\gamma(T-1)}\sum_{m=0}^{T-t}e^{-\alpha\gamma m} \\ &\leq \frac{2D}{1-e^{-\alpha\gamma}}e^{-\alpha\gamma(T-1)}\end{aligned}$$

we can upper bound Δ_2 by

$$\begin{aligned}\Delta_2 &= C_A \|B_T^\top B_T \nabla_{\hat{w}^*} f\| \|\nabla_{\hat{w}^*} f\| \sum_{k=t+1}^{T-1} \|w_{T-1} - w_{k-1}\| \|A_{t+1} \cdots A_{k-1}\| \|A_T\|^{T-k} \\ &= \|B_T^\top B_T \nabla_{\hat{w}^*} f\| \|\nabla_{\hat{w}^*} f\| \times \frac{2C_A D}{1 - e^{-\alpha\gamma}} e^{-\alpha\gamma(T-1)}\end{aligned}$$

Bound on Δ_3 Because

$$\|A_k - (1-\gamma\alpha)I\| = \|\gamma\left(\alpha I - \nabla_w^2 f(w_{k-1})\right)\| \leq \gamma(\beta - \alpha)$$

we can upper bound Δ_3 by

$$\Delta_3 = \|\nabla_{\hat{w}^*} f\| \|B_T^\top B_T \nabla_{\hat{w}^*} f\| \|A_t - (1-\gamma\alpha)I\|^{T-t} \leq \|\nabla_{\hat{w}^*} f\| \|B_T^\top B_T \nabla_{\hat{w}^*} f\| \left(\gamma(\beta-\alpha)\right)^{T-t}$$

Final Result Using the bounds on Δ_1, Δ_2, and Δ_3, we prove the final result.

$$\begin{aligned}&\left(B_T \nabla_{\hat{w}^*} f\right)^\top B_t A_{t+1} \cdots A_T \nabla_{\hat{w}^*} f \\ &\geq (1-\gamma\alpha)^{T-t} \|B_T \nabla_{\hat{w}^*} f\|^2 - \Delta_1 - \Delta_2 - \Delta_3 \\ &\geq (1-\gamma\alpha)^{T-t} \|B_T \nabla_{\hat{w}^*} f\|^2 - \|\nabla_{\hat{w}^*} f\|^2 O\left(\frac{e^{-\alpha\gamma(T-1)}}{1-e^{-\alpha\gamma}} + \left(\gamma(\beta-\alpha)\right)^{T-t}\right)\end{aligned}$$

because B_T has full column rank and

$$\begin{aligned}\Delta_1 + \Delta_2 + \Delta_3 &\leq \|\nabla_{\hat{w}^*} f\|^2 \left(\frac{2C_A D}{1-e^{-\alpha\gamma}} e^{-\alpha\gamma(T-1)} + 2C_B D e^{-\alpha\gamma(T-1)} + \left(\gamma(\beta-\alpha)\right)^{T-k}\right) \\ &= \|\nabla_{\hat{w}^*} f\|^2 \times O\left(\frac{e^{-\alpha\gamma(T-1)}}{1-e^{-\alpha\gamma}} + \left(\gamma(\beta-\alpha)\right)^{T-t}\right)\end{aligned}$$

■

A.3 Proof of Theorem 2.3.3

Theorem 2.3.3. *Suppose F is smooth and bounded below, and suppose there is $\epsilon < \infty$ such that $\|h_{T-K} - \mathrm{d}_\lambda f\| \leq \epsilon$. Using h_{T-K} as a stochastic first-order oracle with a decaying step*

size $\eta_\tau = O(1/\sqrt{\tau})$ to update λ with gradient descent, it follows after R iterations,

$$\mathbb{E}\left[\sum_{\tau=1}^{R} \frac{\eta_\tau \|\nabla F(\lambda_\tau)\|^2}{\sum_{\tau=1}^{R} \eta_\tau}\right] \leq \widetilde{O}\left(\epsilon + \frac{\epsilon^2 + 1}{\sqrt{R}}\right).$$

That is, under the assumptions in Proposition 2.3.1, learning with h_{T-K} converges to an ϵ-approximate stationary point, where $\epsilon = O((1-\gamma\alpha)^{-K})$.

Proof. The proof of this theorem is a standard proof of non-convex optimization with biased gradient estimates. Here we include it for completeness, as part of it will be used later in the proof of Theorem 2.3.4.

Let λ_τ be the τth iterate. For short hand, we write $\mathrm{d}_\lambda f_{(\tau)} = \mathrm{d}_\lambda f(\lambda_\tau)$, and $h_{T-K,(\tau)} = h_{T-K}(\lambda_\tau)$. Assume F is L-smooth and $\|\mathrm{d}_\lambda f_{(\tau)}\| \leq G$ and $\|h_{T-K,(\tau)}\| \leq G$ almost surely for all τ. Then by L-smoothness, it satisfies

$$F(\lambda_{\tau+1}) \leq F(\lambda_\tau) + \langle \nabla F(\lambda_\tau), \lambda_{\tau+1} - \lambda_\tau \rangle + \frac{L}{2}\|\lambda_{\tau+1} - \lambda_\tau\|^2.$$

Let $e_\tau = \mathrm{d}_\lambda f_{(\tau)} - h_{T-K,(\tau)}$ be the error in the gradient estimate. Substitute the recursive update $\lambda_{\tau+1} = \lambda_\tau - \eta_t h_{T-K,(\tau)}$ to the above inequality. Conditioned on λ_τ, it satisfies

$$\mathbb{E}_{|\lambda_\tau}[F(\lambda_{\tau+1})] \leq F(\lambda_\tau) + \mathbb{E}_{|\lambda_\tau}\left[-\eta_t \langle \nabla F(\lambda_\tau), h_{T-K,(\tau)} \rangle + \frac{L\eta_t^2}{2}\|h_{T-K,(\tau)}\|^2\right].$$

Because

$$\begin{aligned} -\mathbb{E}_{|\lambda_\tau}[\langle \nabla F(\lambda_\tau), h_{T-K,(\tau)} \rangle] &= \mathbb{E}_{|\lambda_\tau}\left[-\langle \nabla F(\lambda_\tau), \mathrm{d}_\lambda f_{(\tau)} \rangle + \langle \nabla F(\lambda_\tau), e_\tau \rangle\right] \\ &\leq -\|\nabla F(\lambda_\tau)\|^2 + G\|e_\tau\| \end{aligned} \tag{A.2}$$

and

$$\frac{1}{2}\|h_{T-K,(\tau)}\|^2 = \frac{1}{2}\|\mathrm{d}_\lambda f_{(\tau)}\|^2 + \frac{1}{2}\|e_\tau\|^2 - \langle \mathrm{d}_\lambda f_{(\tau)}, h_{T-K,(\tau)} \rangle \leq \frac{3G^2}{2} + \frac{1}{2}\|e_\tau\|^2$$

we can upper bound $\mathbb{E}_{|\lambda_\tau}[F(\lambda_{\tau+1})]$ as

$$\mathbb{E}_{|\lambda_\tau}[F(\lambda_{\tau+1})] \leq F(\lambda_\tau) + \mathbb{E}_{|\lambda_\tau}\left[-\eta_\tau\|\nabla F(\lambda_\tau)\|^2 + \eta_\tau G\|e_\tau\| + L\eta_\tau^2\left(\frac{3G^2}{2} + \frac{1}{2}\|e_\tau\|^2\right)\right]$$

Performing telescoping sum with the above inequality, we have

$$\begin{aligned}\mathbb{E}\left[\sum_{\tau=1}^{R}\eta_\tau\|\nabla F(\lambda_\tau)\|^2\right] &\leq F(\lambda_1) + \mathbb{E}\left[\sum_{\tau=1}^{R} G\eta_\tau\|e_\tau\| + L\eta_\tau^2\left(\frac{3G^2}{2} + \frac{1}{2}\|e_\tau\|^2\right)\right] \\ &\leq F(\lambda_1) + \sum_{\tau=1}^{R}\left(G\epsilon\eta_\tau + \frac{L(3G^2+\epsilon^2)}{2}\eta_\tau^2\right)\end{aligned}$$

Dividing both sides by $\sum_{\tau=1}^{R}\eta_\tau$ and using the facts that $\eta_\tau = O(\frac{1}{\sqrt{\tau}})$ and that

$$\frac{\sum_{\tau=1}^{R}\frac{1}{\tau}}{\sum_{\tau=1}^{R}\frac{1}{\sqrt{\tau}}} = O\left(\frac{\log R}{\sqrt{R}}\right)$$

proves the theorem. ■

A.4 Proof of Theorem 2.3.4

Theorem 2.3.4. *Under the assumptions in Proposition 2.3.1 and Theorem 2.3.3, if in addition*

1. *g is second-order continuously differentiable*
2. *B_t has full column rank around w_T*
3. *$\nabla_\lambda f^\top(\mathrm{d}_\lambda f + h_{T-K} - \nabla_\lambda f) \geq \Omega(\|\nabla_\lambda f\|^2)$*
4. *the problem is deterministic (i.e. $F = f$)*

then for all $K \geq 1$, with T large enough and γ small enough, the limit point is an exact stationary point, i.e. $\lim_{\tau\to\infty}\|\nabla F(\lambda_\tau)\| = 0$.

Proof. First we consider the special case when S is deterministic. Let $H \geq K$. We

decompose the full gradients into four parts

$$\nabla F = \mathrm{d}_\lambda f = \nabla_\lambda f + q + r + e$$

where

$$\begin{aligned} q &= \sum_{t=T-K+1}^{T} B_t A_{t+1} \cdots A_T \nabla_{\hat{w}^*} f \\ r &= \sum_{t=T-H+1}^{T-K} B_t A_{t+1} \cdots A_T \nabla_{\hat{w}^*} f \\ e &= \sum_{t=0}^{T-H} B_t A_{t+1} \cdots A_T \nabla_{\hat{w}^*} f \end{aligned}$$

We assume that w_t enters a locally strongly convex region for $t \geq H$. This implies, by Proposition 2.3.1, that $\|e\| \leq O(e^{-\alpha\gamma H}\|\nabla_{\hat{w}^*} f\|)$.

To prove the theorem, we first verify two conditions:

1. By Lemma 2.3.2, the assumption $\nabla_\lambda f^\top (\mathrm{d}_\lambda f + h_{T-K} - \nabla_\lambda f) \geq \Omega(\|\nabla_\lambda f\|^2)$, and $\|e\| \leq O(e^{-\alpha\gamma H}\|\nabla_{\hat{w}^*} f\|)$:

$$\begin{aligned} &\mathrm{d}_\lambda f^\top h_{T-K} = (\nabla_\lambda f + q + r + e)^\top (\nabla_\lambda f + q) \\ &= \|\nabla_\lambda f\|^2 + \nabla_\lambda f^\top (q + e + r) + q^\top \nabla_\lambda f + q^\top (q + r) + q^\top e \\ &\geq \Omega(\|\nabla_\lambda f\|^2) + q^\top (q + r) + q^\top e && \text{(Assumption)} \\ &\geq \Omega(\|\nabla_\lambda f\|^2) + \Omega(\|\nabla_{\hat{w}^*} f\|^2) + q^\top e && \text{(Lemma 2.3.2)} \\ &\geq \Omega(\|\nabla_\lambda f\|^2) + \Omega(\|\nabla_{\hat{w}^*} f\|^2) - O\left(e^{-\alpha\gamma H}\|\nabla_{\hat{w}^*} f\|^2\right) && (\|e\| \leq O(e^{-\alpha\gamma H}\|\nabla_{\hat{w}^*} f\|)) \end{aligned}$$

where we note

$$\begin{aligned} \mathrm{d}_\lambda f + h_{T-K} - \nabla_\lambda f &= \nabla_\lambda f + q + r + e + \nabla_\lambda f + q - \nabla_\lambda f \\ &= \nabla_\lambda f + q + r + e + q \end{aligned}$$

Therefore, for H large enough, it holds that

$$\mathrm{d}_\lambda f^\top h_{T-K} \geq \Omega(\|\nabla_\lambda f\|^2 + \|\nabla_{\hat{w}^*} f\|^2) \tag{A.3}$$

2. By definition of $h_{T-K} = \nabla_\lambda f + q$, it holds that

$$\|h_{T-K}\|^2 \leq 2\|\nabla_\lambda f\|^2 + 2\|q\|^2 \leq O(\|\nabla_\lambda f\|^2 + \|\nabla_{\hat{w}^*} f\|^2) \tag{A.4}$$

Next, we prove a lemma

Lemma A.4.1. *Let f be a lower-bound and L-smooth function. Consider the iterative update rule*

$$x_{t+1} = x_t - \eta g_t$$

where g_t satisfies $g_t^\top \nabla f(x_t) \geq c_1 h_t^2$ and $\|g_t\|^2 \leq c_2 h_t^2$, for some constant $c_1, c_2 > 0$ and scalar h_t. Suppose f is lower-bounded and η is chosen such that $\left(-c_1\eta + \frac{Lc_2\eta^2}{2}\right) \leq 0$. Then $\lim_{t\to\infty} h_t = 0$.

Proof. By L-smoothness,

$$\begin{aligned} f(x_{t+1}) - f(x_t) &\leq \nabla f(x_t)^\top (x_{t+1} - x_t) + \frac{L}{2}\|x_{t+1} - x_t\|^2 \\ &= -\eta \nabla f(x_t)^\top g_t + \frac{L\eta^2}{2}\|g_t\|^2 \\ &\leq \left(-c_1\eta + \frac{Lc_2\eta^2}{2}\right) h_t^2 \end{aligned}$$

By telescoping sum, we can show $\sum_{t=0}^{\infty} \left(c\eta - \frac{L\eta^2}{2}\right) h_t^2 < \infty$, which implies $\lim_{t\to\infty} h_t = 0$. ■

Finally, we prove the main theorem by applying Lemma A.4.1. Consider a deterministic problem. Take $h_t^2 = \|\nabla_\lambda f(\lambda_t)\|^2 + \|\nabla_{\hat{w}^*} f(\lambda_t)\|^2$. Because of (A.3) and (A.4), by

Lemma A.4.1, it satisfies that

$$\lim_{t\to\infty} h_t = \lim_{t\to\infty} \|\nabla_\lambda f(\lambda_t)\|^2 + \|\nabla_{\hat{w}^*} f(\lambda_t)\|^2 = 0$$

As $\|\mathrm{d}_\lambda f\| \leq O(\|\nabla_\lambda f\| + \|\nabla_{\hat{w}^*} f\|)$, it shows $\|\mathrm{d}_\lambda f\|$ converges to zero in the limit.

■

A.5 Proof of Theorem 2.3.5

Theorem 2.3.5. *There is a problem, satisfying all but assumption 3 in Theorem 2.3.4, such that optimizing λ with h_{T-K} does not converge to a stationary point.*

Proof. We prove the non-convergence using the following strategy. First we show that, when assumption 3 in Theorem 2.3.4, i.e.

$$\nabla_\lambda f^\top (\mathrm{d}_\lambda f + h_{T-K} - \nabla_\lambda f) \geq \Omega(\|\nabla_\lambda f\|^2) \tag{A.5}$$

does not hold, there is some problem such that $h_{T-k} \neq 0$ for all stationary points (i.e. λ such that $\mathrm{d}_\lambda f = 0$). Then we show that, for such a problem, optimizing λ with h_{T-k} cannot converge to any of the stationary points.

Counter example To construct the counterexample, we consider a scalar deterministic bilevel optimization problem of the form

$$\begin{aligned} &\min_\lambda \frac{1}{2}(\hat{w}^*)^2 + \phi(\lambda) \\ \text{s.t.} \quad &\hat{w}^* \approx w^* \in \arg\min_w \frac{1}{2}(w-\lambda)^2 \end{aligned} \tag{A.6}$$

in which ϕ is some perturbation function that we will later define, and $\hat{w}^*$ is computed by performing $T > 1$ steps of gradient descent in the lower-level optimization problem with

some constant initial condition w_0 and constant step size $0 < \gamma < 1$, i.e.

$$\hat{w}^* = w_T, \qquad w_{t+1} = w_t - \gamma(w_t - \lambda)$$

We can observe this problem satisfies *almost* all the assumptions in Theorem 2.3.4:

1. The lower-level objective g is smooth and strongly convex. (Proposition 2.3.1)
2. The upper-level objective F is smooth. (Theorem 2.3.3)
3. The lower-level objective g is second-order continuously differentiable (assumption 1 in Theorem 2.3.4)
4. The Jacobian if full rank, i.e. $B_t = \gamma > 0$ (assumption 2 in Theorem 2.3.4)
5. The upper-level objective function is deterministic, i.e. $F = f$ (assumption 4 in Theorem 2.3.4)

But we will show that properly setting ϕ can break the non-interfering assumption in (A.5) (i.e. assumption 3 in Theorem 2.3.4) and then creates a problem such that optimizing λ with K-RMD does not converge to an exact stationary point.

We follow the two-step strategy mentioned above.

Step 1: Non-vanishing approximate gradient Without loss of generality, let us consider optimizing λ with 1-RMD. In this case we can write the approximate and the exact gradients in closed form as

$$h_{T-1} = \nabla\phi + w^*\gamma, \qquad \mathrm{d}_\lambda f = \nabla\phi + w^*\gamma \sum_{t=0}^{T} (1-\gamma)^{T-t} \tag{A.7}$$

which are given by (2.4) and (2.7). We will show that by properly choosing ϕ, we can define $f(\lambda) = \frac{1}{2}(\hat{w}^*)^2 + \phi(\lambda)$ such that, at any of the stationary points of f, the approximate gradient of 1-RMD does not vanish. That is, we show when $\mathrm{d}_\lambda f = 0$, $h_{T-1} \neq 0$.

Before proceeding, let us define $u = w^*\gamma$ and $v = w^*\gamma \sum_{t=0}^{T}(1-\gamma)^{T-t}$ for convenience. To show how to construct ϕ, let us consider the stationary points in the case[1] when $\phi = 0$. Let P_0 denote the set of these stationary points, i.e. $P_0 = \{\lambda : v = 0\}$. Since f is smooth and lower-bounded, we know that P_0 is non-empty, and from the construction of our counterexample we know that P_0 contains exactly the λs such that $w^* = 0$.

This implies that for $\lambda \in \mathbb{R}\backslash P_0$, it satisfies $w^* \neq 0$ and therefore

$$uv = (w^*\gamma)^2 \sum_{t=0}^{T}(1-\gamma)^{T-t} > 0 \tag{A.8}$$

We use this fact to pick an adversarial ϕ. Consider any smooth, lower-bounded ϕ whose stationary points are not in P_0, e.g. $\phi(\lambda) = \frac{1}{2}(\lambda - \lambda_0)^2$ and $\lambda_0 \notin P_0$. Then $f(\lambda) = \frac{1}{2}(\hat{w}^*)^2 + \phi(\lambda)$ has a non-empty set of stationary points P_ϕ such that $P_\phi \cap P_0 = \emptyset$. We see that, for such ϕ, the non-interfering assumption (assumption 3 in Theorem 2.3.4) is violated in P_ϕ:

$$\begin{aligned}
\nabla_\lambda f^\top(\mathrm{d}_\lambda f + h_{T-1} - \nabla_\lambda f) &= \nabla_\lambda f^\top(\nabla_\lambda f + u - \nabla_\lambda f) && \because \mathrm{d}_\lambda f = 0 \text{ and } h_{T-1} = \nabla_\lambda f + u \\
&= \nabla_\lambda \phi^\top u \\
&= -vu && \because 0 = \mathrm{d}_\lambda f = \nabla_\lambda \phi + v \\
&< 0 && \because \text{(A.8) and } P_\phi \cap P_0 = \emptyset \\
&< (\nabla_\lambda \phi)^2 && \because v > 0 \text{ for } \lambda \in P_\phi
\end{aligned}$$

And we show for any $\lambda \in P_\phi$ it holds that $h_{T-1} \neq 0$. This can be seen from the definition

$$h_{T-1} = \nabla\phi + u = \mathrm{d}_\lambda f + u - v = u - v \neq 0$$

where the last inequality is because $w^* \neq 0$ for $\lambda \in P_\phi$.

[1]Note in this special case, assumption 3 in Theorem 2.3.4 holds trivially when $\phi(\lambda) = 0$ (i.e. $\nabla_\lambda f = 0$) and optimizing λ with K-RMD converges to an exact stationary point.

Step 2: Non-convergence to any stationary point We have shown that there is a problem which satisfies all the assumptions but assumption 3 of Theorem 2.3.4, and at any of its stationary points (i.e. when $\mathrm{d}_\lambda f = 0$) we have $h_{T-K} \neq 0$. Now we show this property implies failure to converge to the stationary points for the general problems considered in Theorem 2.3.5 (i.e. we do not rely on the form made in Step 1 anymore).

We prove this by contradiction. Let λ^* be one of the stationary points. We choose $\delta_0 > 0$ such that, for some $\epsilon > 0$, $\|h_{T-K}\| > \epsilon/\gamma$ for all λ inside the neighborhood $\{\lambda : \|\lambda - \lambda^*\| < \frac{\delta_0}{2}\}$, where we recall γ is the step size of the lower-level optimization problem. A non-zero δ_0 exists because h_{T-1} is continuous by our assumption and $h_{T-K} \neq 0$ at λ^*.

We are ready to show the contradiction. Let $\delta = \min\{\delta_0, \epsilon\}$. Suppose there is a sequence $\{\lambda_\tau\}$ that converges to the stationary point λ^*. This means that there is $0 < M < \infty$ such that, $\forall \tau \geq M$, $\|\lambda_\tau - \lambda^*\| < \frac{\delta}{2}$, which implies that $\forall \tau \geq M$, $\|\lambda_{\tau+1} - \lambda_\tau\| < \delta$. However, by our choice of δ_0, $\|\lambda_{\tau+1} - \lambda_\tau\| = \gamma\|h_{T-K}\| > \epsilon \geq \delta$, leading to a contradiction.

Thus, no sequence $\{\lambda_\tau\}$ converges to any of the stationary points. This concludes our proof. ■

A.6 Proof of Proposition 2.3.6

Proposition 2.3.6. *Under the assumptions in Proposition 2.3.1, suppose w_t converges to a stationary point w^*. Let $A_\infty = \lim_{t\to\infty} A_t$ and $B_\infty = \lim_{t\to\infty} B_t$. For $\gamma < \frac{1}{\beta}$, it satisfies that*

$$-\nabla_{\lambda,w} g \nabla_{w,w}^{-1} g = B_\infty \textstyle\sum_{k=0}^{\infty} A_\infty^k \tag{2.11}$$

Proof. Recall our shorthand that $\nabla_{\lambda,w} g$ and $\nabla_{w,w} g$ are evaluated at (w^*, λ). In the limit, it

holds that

$$\lim_t A_t = \lim_t \nabla_w \Xi_t(w_{t-1}, \lambda) = \nabla_w(w^* - \gamma \nabla_w g(w^*, \lambda)) = I - \gamma \nabla_{w,w} g =: A_\infty$$
$$\lim_t B_t = \lim_t \nabla_\lambda \Xi_t(w_{t-1}, \lambda) = \nabla_\lambda(w^* - \gamma \nabla_w g(w^*, \lambda)) = -\gamma \nabla_{\lambda,w} g =: B_\infty$$

To prove the equality (2.11), we use Lemma (A.6.1).

Lemma A.6.1. *Horn and Johnson, 1990 For a matrix A with $\|A\| < 1$, it satisfies that*

$$(I - A)^{-1} = \sum_{k=0}^{\infty} A^k$$

Since $\gamma \leq \frac{1}{\beta}$, we have $\gamma \alpha I \preceq \gamma \nabla_{w,w} g \preceq I$, so $\|I - \gamma \nabla_{w,w} g\| < 1$. By Lemma A.6.1,

$$\nabla_{w,w}^{-1} g = \gamma (I - I + \gamma \nabla_{w,w} g)^{-1} = \gamma \sum_{k=0}^{\infty} (I - \gamma \nabla_{w,w} g)^k = \gamma \sum_{k=0}^{\infty} A_\infty^k$$

Therefore,

$$-\nabla_{\lambda,w} g \nabla_{w,w}^{-1} g = (-\gamma \nabla_{\lambda,w} g) \left(\frac{1}{\gamma} \nabla_{w,w}^{-1} g \right) = B_\infty \sum_{k=0}^{\infty} A_\infty^k$$

■

A.7 Detailed experimental setup

In this appendix, we provide more details about the settings we used in each experiment. We use Adam Kingma and Ba, 2015 to optimize the upper-level objective and vanilla gradient descent for the lower objective. We denote by $\hat{w}^*$ the results of running T steps of gradient descent with step size γ.

A.7.1 Data hypercleaning

In this appendix, we provide more details about the data hypercleaning experiment on MNIST from Section 2.4.2.

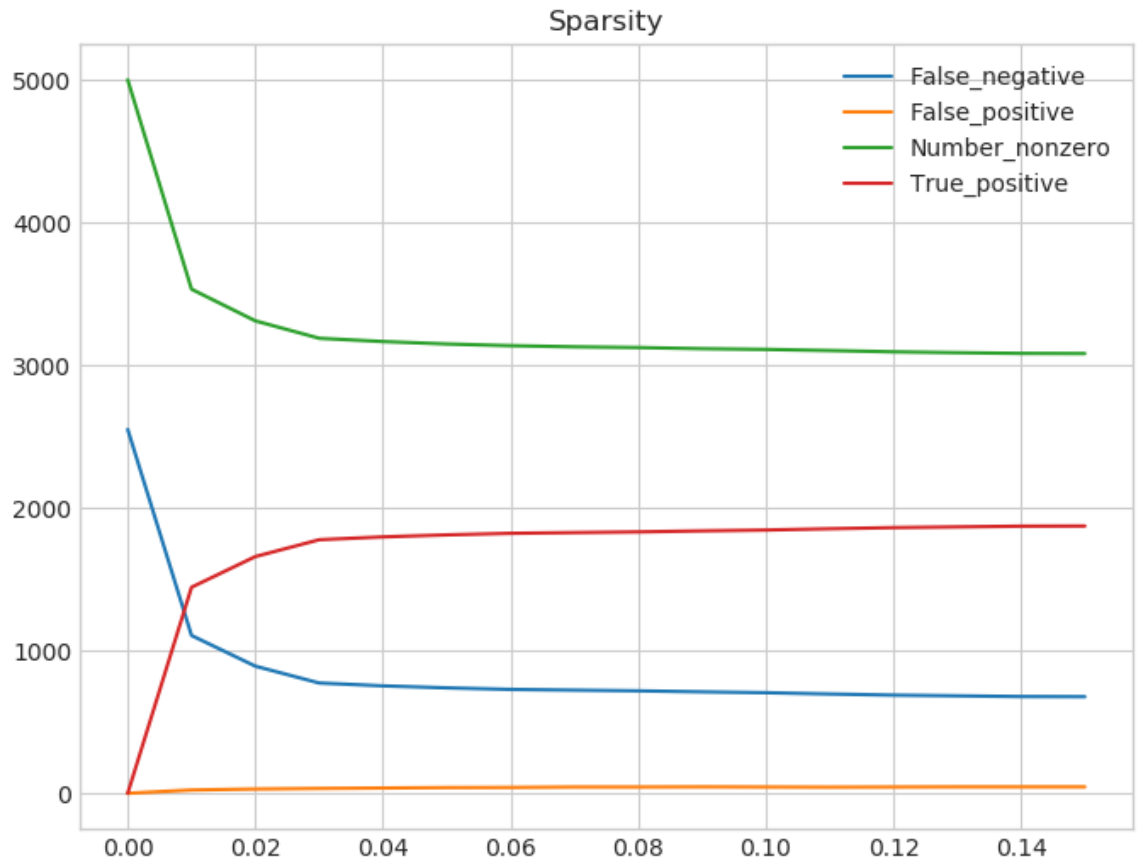
Both the training and the validation sets consist of 5000 class-balanced examples from the MNIST dataset. The test set consists of the remaining examples. For each training example, with probability $\frac{1}{2}$, we replaced the label with a uniformly random one.

For various K, we performed K-RMD for 1000 hyperiterations. Like in the toy experiment (Section 2.4.1) we adjusted the initial meta-learning rate η_0 for each K so that the norm of the initial update was roughly the same for each K.

We asserted earlier that the reported F1 scores are not sensitive to our choice of threshold $\lambda_i < -3$. To validate this assertion, we repeated the experiment for various thresholds. F1 scores are reported in the table below.

K	$\lambda_i < -4$	$\lambda_i < -3$	$\lambda_i < -1$
1	0.84	0.84	0.84
5	0.89	0.89	0.90
25	0.89	0.89	0.89
50	0.89	0.89	0.89
100	0.89	0.89	0.89

We only ran these experiments for 150 hyperiterations, because the F1 score has essentially converged by that point. Indeed, the plot below shows identification of corrupted labels for $K = 1$, with cutoff $\lambda_i < -4$. The X axis is in units of 1000 hyperiterations. We see that 1-RMD rapidly identifies most of the mislabeled examples, with a few false positives.



A.7.2 Task interaction

We use $T = 100$ iterations of gradient descent with learning rate 0.1 in the lower objective which yields $\hat{w}_S^*$. To ensure that C is symmetric, and that C_{ij} and ρ are nonnegative, we re-parametrize them as $\rho = \text{softplus}(\nu)$ and $C = A + A^\top$, where $A_{ij} = \text{softplus}(B_{ij})$ and B is a hyperparameter matrix. Thus, the hyperparameters to be optimized are $\lambda = \{B, \nu\}$.

Rather than using raw pixels, we extract image features from the output of the average pooling layer in Resnet-18 He et al., 2016 which is trained on ImageNet Deng et al., 2009b. We use the same data pre-processing that is used for training Resnet architecture.

When reporting test accuracy, we run 10 independent trials. In each trial, we sample the training and validation datasets with a balanced set of m examples each ($m = 50$ for CIFAR-10 and $m = 300$ for CIFAR-100) and use the rest of the dataset for testing. To avoid over-fitting, we use early stopping when the testing error does not improve for 500

hyper-iterations.

Although we are using a similar setting as Franceschi et al., 2017b, our results on full back-propagation are quite different from theirs. We believe it is because we are using a different network architecture and pre-processing method for feature extraction.

A.7.3 One-shot classification

Dataset The Omniglot dataset Lake, Salakhutdinov, and Tenenbaum, 2015, a popular benchmark for few-shot learning, is used in this experiment. We consider 5-way classification with 1 training and 15 validation examples for each of the five classes. To evaluate the generalization performance, we restrict the meta-training dataset to a random subset of 1200 of the 1623 Omniglot characters. The meta-validation dataset consists of 100 other characters, and meta-testing dataset has the remaining 323 characters. We use the meta-validation dataset for tuning the upper-level optimization parameters and report the performance of the algorithm on the meta-testing dataset. Note that no data augmentation method is used in the training.

Neural Network and Optimization The overall neural network architecture is shown in Figure A.1. Our architecture inherits the hyper-representation model of Franceschi et al., 2017a with some modifications. The first two convolutional layers, parametrized by hyperparameter $\lambda = \{\lambda_{l_1}, \lambda_{l_2}\}$, transform the input image into a "hyper-representation" space. The last three layers, parametrized by $w = \{w_{l_3}, w_{l_4}, w_{l_5}\}$ are fine-tuned in the lower-level optimization. Additionally, we have regularization hyperparameters $\lambda_r = \{\rho_i\}_{i=1}^3 \cup \{c_j\}_{j=1}^3$. The overall setup corresponds essentially to meta-learning the two bottom layers of a CNN; for each task, the weights in the first two layers are frozen, and the k-way classifier of the last three layers is fine tuned. Overall, the model has ≈ 110k hyperparameters and ≈ 75k parameters.

We use a meta-batch-size of 4 in each hyper-iteration. To limit the training time, we

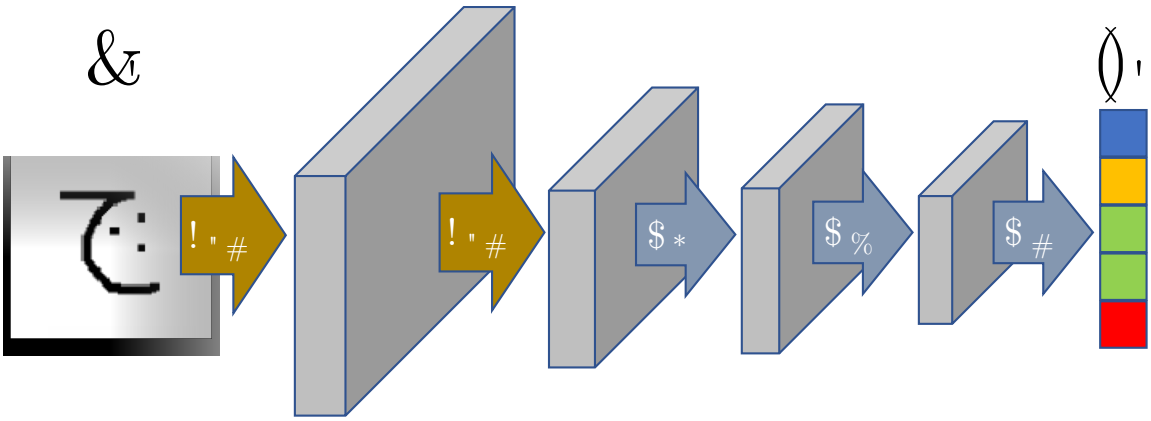


Figure A.1: One-shot learning network architecture. The first two convolutional layers map the input image into a "hyper-representation" space which is frozen while optimizing the lower-level objective. The last three layers are tuned for each task and regularized to avoid overfitting. All the convolutional layers have 64 3×3 kernels. There is a max-pooling layer followed by a batch-normalization and a ReLU layer after each convolution.

stop all the algorithms after 5000 hyper-iterations. Needless to say, these results could be further improved by using data augmentation, higher meta-batch size, and running more hyper-iterations. However, our current setup is selected so that all the experiments can be run in a reasonable amount of time, while sharing a similar setting used in practical one-shot learning.

APPENDIX B

ONE-SHOT LEARNING FOR SEMANTIC SEGMENTATION

B.1 Weight Hashing

In 3.3, we employed the weight hashing technique from Chen et al., 2015 to map the 1000-dimensional vector output from the last layer of VGG to the 4097 dimensions of $\{w, b\}$. This mapping (1) reduces the variance of $\{w, b\}$ as was also noted by Huh *et al*. in Noh, Hongsuck Seo, and Han, 2016, and (2) reduces the overfitting which would occur due to the massive number of extra parameters that a fully connected layer will introduce if used instead.

Weight hashing is explained as decompression in Chen et al., 2015 and is performed as follows. Let $x \in \mathbb{R}^m$ and $\theta \in \mathbb{R}^d$, typically $d > m$, be the inputs and outputs of the layer respectively. Weight hashing works by replicating each coefficient of x in multiple locations of θ and randomly flipping the sign to reduce the covariance of copied coefficients. Illustrated in Figure B.1. Specifically, the i^{th} coefficient of θ is

$$\theta(i) = x(p)\zeta(i), \tag{B.1}$$

$$p = \kappa(i), \tag{B.2}$$

where both $\kappa(i) \rightarrow \{1, \ldots, m\}$ and $\zeta(i) \rightarrow \{-1, +1\}$ are hashing functions determined randomly. While Chen et al., 2015 perform the hashing implicitly to keep the memory footprint small, we implement it as a fully connected layer since we can keep both the

Figure B.1: Illustration of weight hashing. In the figure, x is mapped to y by replicating coefficients of x in multiple random locations of y and randomly flipping the sign. The colors help indicate where the entries are copied from.

hashing values and θ in memory. The weights are set as

$$W(i,j) = \zeta(i)\delta_j(\kappa(i)), \tag{B.3}$$

where $\delta_j(\cdot)$ is discreet Dirac delta function. The weights are set according to the random hashing functions before training and are kept fixed. This is both easier to implement and more computationally efficient than the original formulation and that used by Noh, Hongsuck Seo, and Han, 2016. The output of the inner product layer Wx is equal to θ from Equation B.1.

B.2 Siamese Network for Dense Matching

We used the adapted version of Siamese Neural Network for One-shot Image Recognition by Koch *et al*. Koch, 2015 for one-shot image segmentation. Here we explain the implementation details. The method from Koch, 2015 receives as input two images that are each passed through identical convolutional networks and produce a vector as the output for each of them. These vectors are then compared using a learned $L1$ similarity metric and the image is classified according to the label of its nearest neighbor in this metric space. In our case, we use an FCN that outputs a feature volume each for both query and support images. Then the feature for every pixel in the query image is compared to every pixel in the support using a learned $L1$ similarity metric. We implemented this cross similarity measurement between pixels as a python layer for Caffe. The binary label here is assigned to each pixel according to the label of the nearest pixel label in the support set. The whole structure is illustrated in Figure B.2.

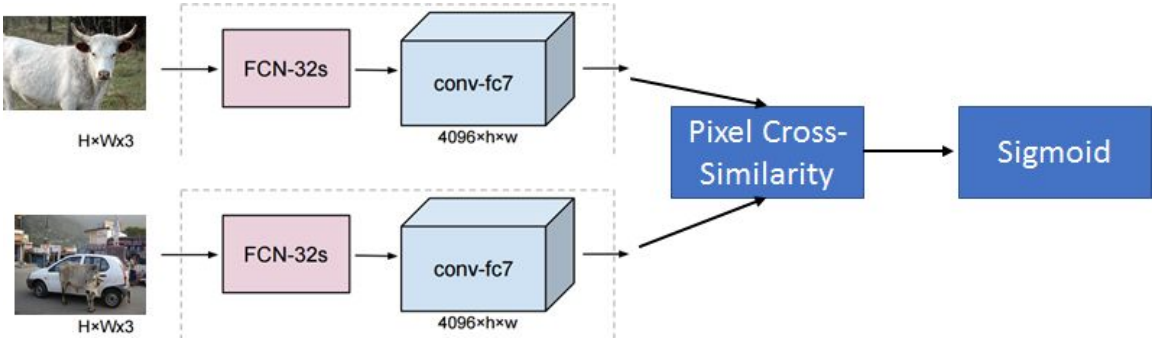


Figure B.2: Siamese network architeture for dense matching.

During training, we use FCNs initialized on ImageNet Deng et al., 2009a and at each iteration we sample a pair of images from the PASCAL-5^i training set. One of them is treated as the query image and the other becomes the support image. The gradient is computed according to the cross-entropy loss between the sigmoid of the similarity metric and the true binary label. Every batch contains a subset (50%) of the pixels of a query and a support image. Both the similarity metric and the FCN feature extraction are jointly as different

parts of the same neural network.

B.3 Qualitative Results

We include some more qualitative results of our approach for One Shot Semantic Segmentation in Figure B.3. We see that our method is capable of segmenting a variety of classes well and can distinguish an object from others in the scene given only a single support image.

We illustrate the effect of conditioning by segmenting the same query image with different support sets in Figure B.4. We picked an unseen query image with two unseen classes, car and cow, and sample support image-mask pairs for each class. Figure B.5 shows how increasing size of the support set helps improve the predicted mask. Note that in both Figures B.5 and B.4 yellow indicates the overlap between ground truth and the prediction.

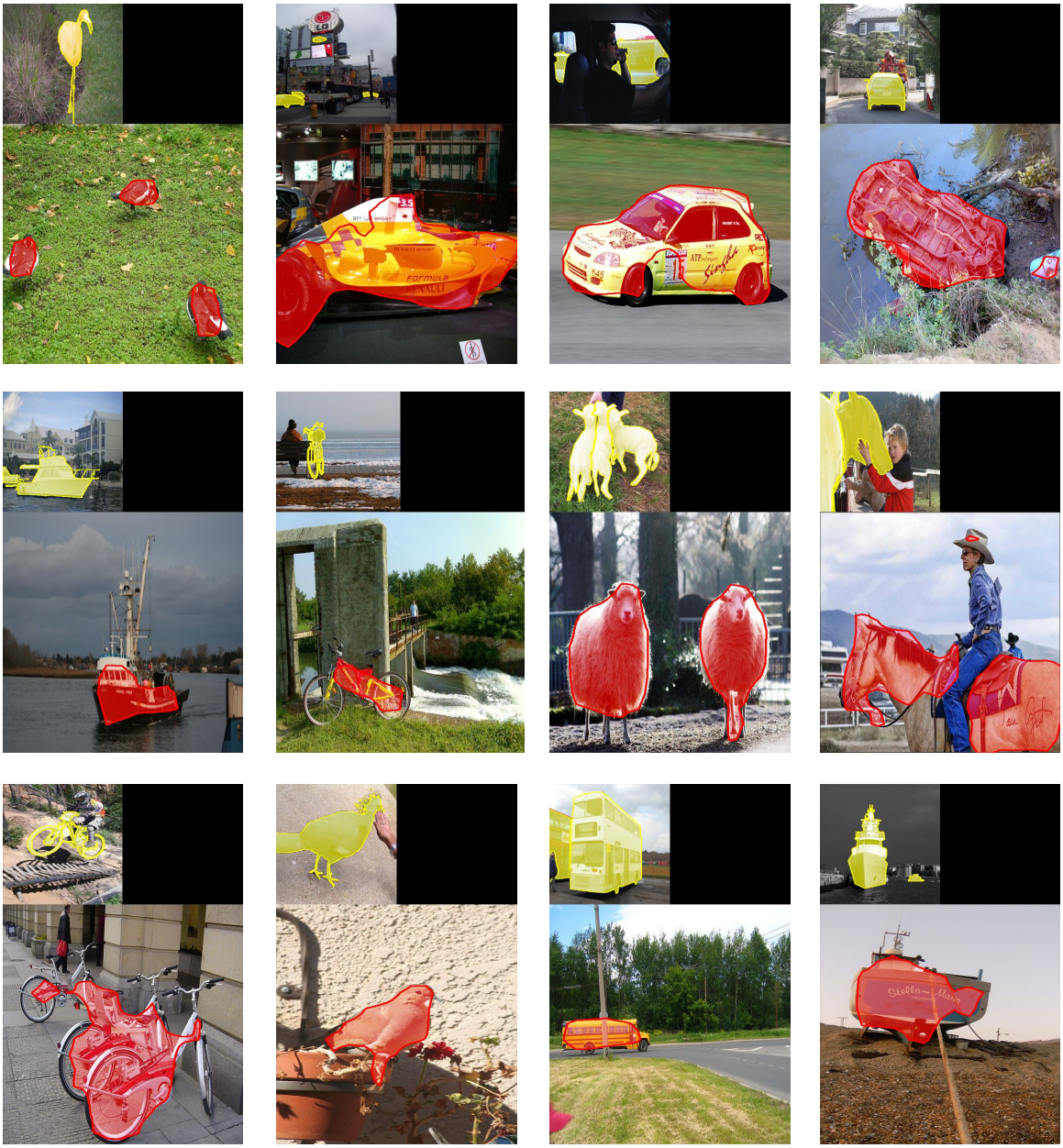


Figure B.3: Qualitative results for 1-shot. Inside each tile, we have the support set at the top and the query image at the bottom. The support is overlaid with the ground truth in yellow and the query is overlaid with our predicted mask in red.

Figure B.4: Illustration of conditioning effect. Given a fix query image, predicted mask changes by changing the support set. Ground-truth mask is shown in green. First row: support image-mask pairs are sampled from cow class. Second row: support image-mask pairs are sampled from car class. First column: only changing the support mask will will change the prediction.

Figure B.5: Effect of increasing the size of the support set. Results of 1-shot and 5-shot learning on the same query image are in the first and second rows respectively. Ground truth masks are shown in green and our prediction is in red. The overlap between ground truth and prediction appears yellow.

APPENDIX C

LEARNING WEAKLY SUPERVISED FEW-SHOT OBJECT DETECTION

C.1 Co-Localization: COCO Dataset Creation and Faster-RCNN Training

COCO dataset has 80 classes in total. We take the same 17 unseen classes which is used in zero-shot object detection paper Bansal et al., 2018 and keep remaining 63 classes for training. The training set is constructed using the images in COCO 2017 train set which contain at least one object from the seen classes. The COCO test set, is built by combining the unused images of the train set and images in COCO validation set which contain at least one object from the unseen classes. Similar to Bansal et al., 2018, to avoid training the network to classify unseen objects as background, we remove objects from unseen classes from the training images using their ground-truth segmentation masks.

We use Tensorflow object-detection API for pre-training the Faster-RCNN feature extraction module Huang et al., 2017. To speed up pre-training, training images are resized down to 336×336 pixels and ResNet-50 He et al., 2016 is used as the backbone feature extractor. All layer weights are initialized with variance scaling initialization Glorot and Bengio, 2010 and biases are set to zero initially. An additional linear layer which maps the 2048 dimensional output of second stage feature extractor to a $d = 640$ dimensional feature vector is added to the network. We did this to have the dimension of the feature space the same as few-shot common object recognition experiment. We pre-train the feature extractor on four GPUs with batch size of 12 for $600k$ iterations. The $d = 640$ dimensional features are used as input to all of the methods in our experiments.

C.2 Hyperparameter Tuning

In the few-shot common object recognition task, we use grid search on the validation set to tune the hyperparameters of all the methods. To ensure that the structured inference methods optimize the same objective function, we find η for the TRWS method and use the same value in AStar and greedy energy functions. For the few-shot common object recognition task value of η is shown in Table C.1 for each setting.

In the Co-Localization experiments, the results of the best performing hyperparameters is reported for all the methods. $\eta = 0.5$ and $\eta = 0.7$ is used in COCO and ImageNet experiments respectively.

C.3 Structured Inference Methods Comparison

Table C.1 compares the performance of different inference algorithms. The success rate of Shaban et al., 2019a greedy method is on par with the other inference algorithms. From the optimization point of view it is also important to see the mean energy value for the top selection of each method. These results are shown in Table C.2 and Table C.3 for few-shot common object recognition and co-localization experiments respectively. While AStar and TRWS achieve lower energy values for this problems, the success rate of the methods are comparable. This suggests that finding an approximate solution for the minimization problem is sufficient for achieving high success rate.

	N	4			8			16		
	B	0	10	20	0	10	20	0	10	20
B = 5	TRWS	$54.55 \pm 1.54(0.0)$	$63.78 \pm 1.49(0.5)$	$65.43 \pm 1.47(0.6)$	$64.55 \pm 1.05(0.0)$	$72.60 \pm 0.98(0.8)$	$73.80 \pm 0.96(1.2)$	$70.29 \pm 0.71(0.0)$	$78.71 \pm 0.63(1.6)$	$80.08 \pm 0.62(1.9)$
	ASTAR	$54.55 \pm 1.54(0.0)$	$63.82 \pm 1.49(0.5)$	$65.48 \pm 1.47(0.6)$	$64.48 \pm 1.05(0.0)$	$72.49 \pm 0.98(0.8)$	$73.99 \pm 0.96(1.2)$	$69.91 \pm 0.71(0.0)$	$78.49 \pm 0.64(1.6)$	$80.03 \pm 0.62(1.9)$
	Greedy(Ours)	$54.55 \pm 1.54(0.0)$	$63.83 \pm 1.49(0.5)$	$65.48 \pm 1.47(0.6)$	$64.48 \pm 1.05(0.0)$	$72.49 \pm 0.98(0.8)$	$73.99 \pm 0.96(1.2)$	$69.67 \pm 0.71(0.0)$	$78.60 \pm 0.64(1.6)$	$79.93 \pm 0.62(1.9)$
B = 10	TRWS	$29.40 \pm 1.41(0.0)$	$37.15 \pm 1.50(0.5)$	$38.50 \pm 1.51(0.7)$	$36.14 \pm 1.05(0.0)$	$42.61 \pm 1.08(0.9)$	$47.59 \pm 1.09(1.1)$	$41.45 \pm 0.76(0.0)$	$50.88 \pm 0.77(1.5)$	$53.71 \pm 0.77(2.3)$
	ASTAR	$29.20 \pm 1.41(0.0)$	$37.43 \pm 1.50(0.5)$	$38.50 \pm 1.51(0.7)$	$35.96 \pm 1.05(0.0)$	$42.83 \pm 1.08(0.9)$	$47.46 \pm 1.09(1.1)$	$41.41 \pm 0.76(0.0)$	$51.32 \pm 0.77(1.5)$	$53.57 \pm 0.77(2.3)$
	Greedy(Ours)	$29.20 \pm 1.41(0.0)$	$37.42 \pm 1.50(0.5)$	$38.50 \pm 1.51(0.7)$	$35.98 \pm 1.05(0.0)$	$42.85 \pm 1.08(0.9)$	$47.63 \pm 1.09(1.1)$	$41.54 \pm 0.76(0.0)$	$51.70 \pm 0.77(1.5)$	$53.63 \pm 0.77(2.3)$

Table C.1: *Success rate of different energy minimization algorithms on miniImageNet. Value of the parameter η is shown in the parenthesis for each experiment. See section 5.2.7 and Table 5.1 for the detailed problem setup.*

	N	4			8			16		
	$\bar{B}$	0	10	20	0	10	20	0	10	20
B = 5	TRWS	2.929179	−4.416873	−4.842334	18.300657	−4.425953	−12.602217	86.034355	−6.873013	−10.020649
	ASTAR	2.908970	−4.429455	−4.851543	18.192284	−4.529052	−12.666497	85.560267	−7.277377	−10.398633
	Greedy	2.908970	−4.429455	−4.851543	18.192282	−4.529052	−12.666499	86.692482	−6.909996	−10.002609
B = 10	TRWS	0.515563	−6.576048	−8.300273	8.749933	−15.959289	−17.238385	53.324193	−28.602048	−59.609459
	ASTAR	0.502832	−6.597286	−8.315386	8.675015	−16.079914	−17.404502	52.819455	−29.388606	−60.499036
	Greedy	0.502832	−6.597286	−8.315387	8.707342	−16.048676	−17.384832	57.168652	−25.869081	−57.885948

Table C.2: *Expected energy for different inference methods. Lower energy is better.*

Method	COCO	ImageNet
TRWS	−28.485636	−28.630786
AStar	−28.487422	−28.631678
Greedy	−27.246355	−25.496649

Table C.3: *Mean energy on COCO and ImageNet with* 8 *positive and* 8 *negative images. Lower energy is better.*

C.4 Sharing Parameters of Unary and Pairwise Relation Modules

As it is discussed in section 5.2.3, both unary and pairwise potential functions use the relation module with an identical architecture. However, since the input class distribution is different for these functions, we choose not to share their parameters. We conduct an experiment to see the effect of parameter sharing in few-shot common object recognition task with $B = 5$, $N = 8$, and $\bar{B} = 10$. As Table 5.1 shows, the success rate for this setting is $72.49 \pm 0.98\%$ without parameter sharing. However, when the unary and pairwise are trained with shared relation module parameters, the performance degrades to $69.35 \pm 1.01\%$.

C.5 More Qualitative Results

Qualitative results on ImageNet dataset are illustrated in Figure C.1. Figure C.2 shows the complete qualitative results shown in Figure 5.3 with the negative images on COCO dataset.

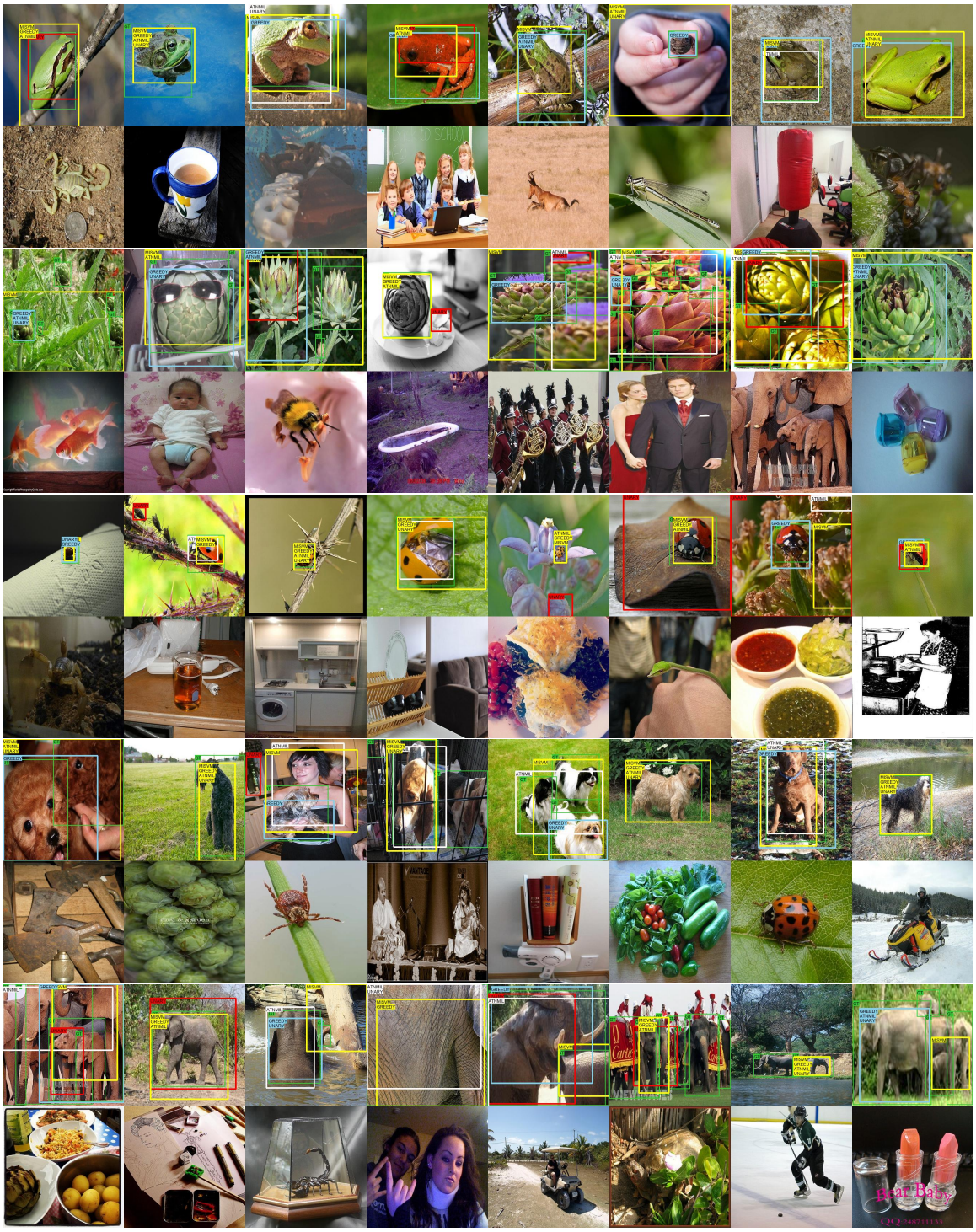


Figure C.1: *Qualitative results on ImageNet dataset. In each problem, the first row and the second row show positive and negative images respectively. While different methods work as good in easier images with one object, the greedy method performs better in harder examples with multiple objects in each image. Selected regions are tagged with method names. Ground-truth target bounding box is shown in green with tag "GT".*

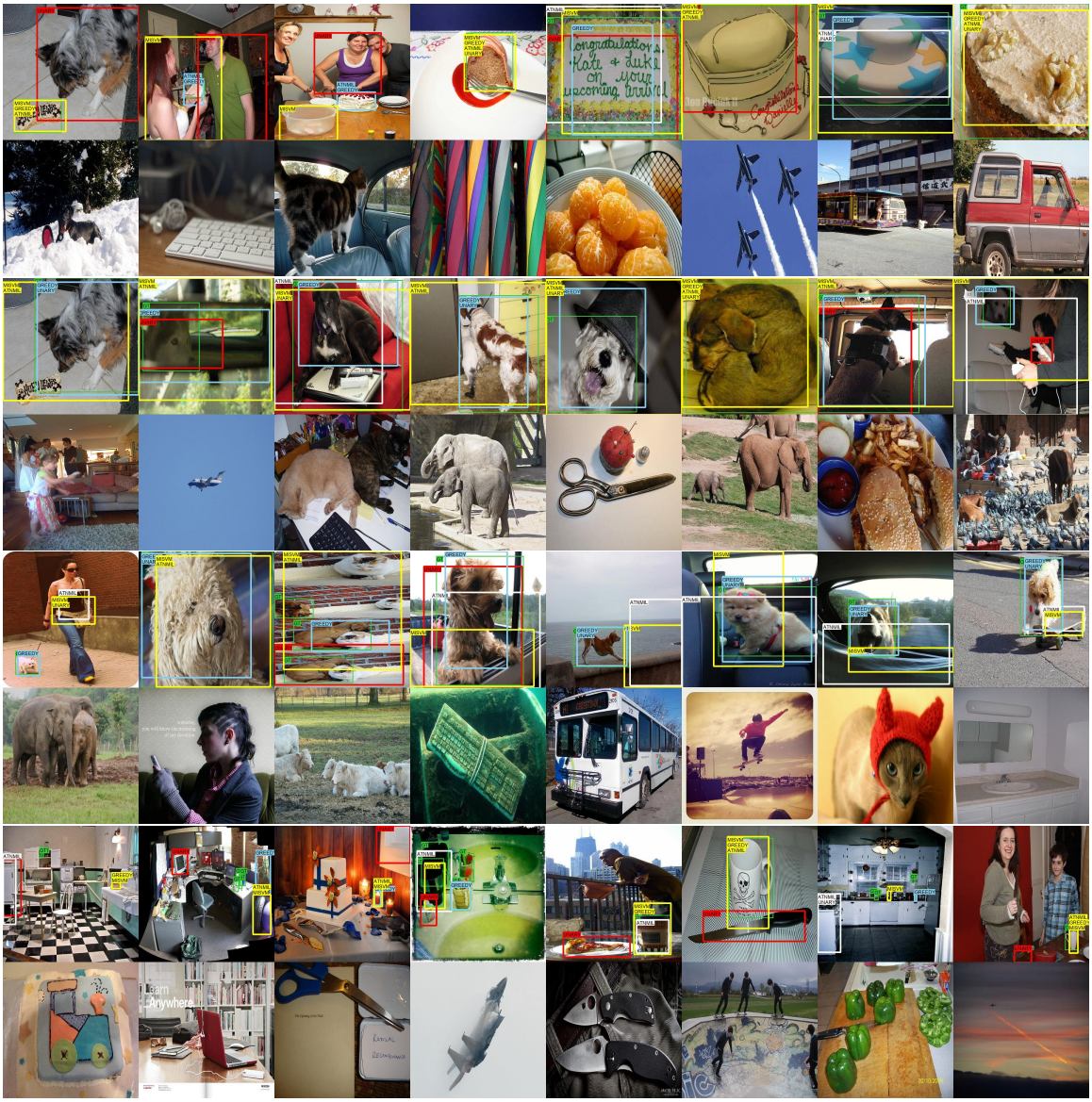


Figure C.2: *Qualitative results on COCO. Complete version of the results shown in Figure 5.3 of the paper with negative images. In the first problem, class "Person" does not appear in the negative images. This could explain why "Unary Only" method detects people in the first problem.*

REFERENCES

Andres, Bjoern, Thorsten Beier, and Joerg H. Kappes (2012). "OpenGM: A C++ Library for Discrete Graphical Models". In: *CoRR* abs/1206.0111.

Andrews, Stuart, Ioannis Tsochantaridis, and Thomas Hofmann (2003). "Support vector machines for multiple-instance learning". In: *NIPS*, pp. 577–584.

Andrychowicz, Marcin, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, and Nando de Freitas (2016). "Learning to learn by gradient descent by gradient descent". In: *Advances in Neural Information Processing Systems*, pp. 3981–3989.

Babenko, Boris, Ming-Hsuan Yang, and Serge Belongie (2009). "Visual tracking with online multiple instance learning". In: *CVPR*. IEEE, pp. 983–990.

Bansal, Ankan, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran (2018). "Zero-shot object detection". In: *ECCV*. http://ankan.umiacs.io/zsd.html, pp. 384–400.

Bart, Evgeniy and Shimon Ullman (2005). "Cross-generalization: Learning novel classes from a single example by feature replacement". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, pp. 672–679.

Batra, Dhruv, Payman Yadollahpour, Abner Guzman-Rivera, and Gregory Shakhnarovich (2012). "Diverse m-best solutions in markov random fields". In: *European Conference on Computer Vision*. Springer, pp. 1–16.

Baydin, Atilim Gunes, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind (2017). "Automatic differentiation in machine learning: A survey". In: *Journal of Machine Learning Research* 18, 153:1–153:43.

Baydin, Atilim Gunes, Robert Cornish, David Martinez Rubio, Mark Schmidt, and Frank Wood (2018). "Online Learning Rate Adaptation with Hypergradient Descent". In: *International Conference on Learning Representations*.

Bengio, Yoshua (2000). "Gradient-based optimization of hyperparameters". In: *Neural computation* 12.8, pp. 1889–1900.

Bergstra, James and Yoshua Bengio (2012). "Random search for hyper-parameter optimization". In: *Journal of Machine Learning Research* 13.Feb, pp. 281–305.

Bergtholdt, Martin, Jörg Kappes, Stefan Schmidt, and Christoph Schnörr (2010). "A study of parts-based object class detection using complete graphs". In: *IJCV* 87.1-2, p. 93.

Bertinetto, Luca, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi (2016). "Learning feed-forward one-shot learners". In: *Advances in Neural Information Processing Systems*, pp. 523–531.

Besag, Julian (1986). "On the statistical analysis of dirty pictures". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 48.3, pp. 259–279.

Bilen, Hakan, Marco Pedersoli, and Tinne Tuytelaars (2015). "Weakly supervised object detection with convex clustering". In: *CVPR*, pp. 1081–1089.

Bunescu, Razvan C. and Raymond J. Mooney (2007). "Multiple Instance Learning for Sparse Positive Bags". In: *ICML*, pp. 105–112.

Caelles, S., K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool (2017a). "One-Shot Video Object Segmentation". In: *Computer Vision and Pattern Recognition (CVPR)*.

Caelles, Sergi, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool (2017b). "One-Shot Video Object Segmentation". In: *Computer Vision and Pattern Recognition*.

Carbonneau, Marc-André, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon (2018). "Multiple instance learning: A survey of problem characteristics and applications". In: *Pattern Recognition* 77, pp. 329–353.

Caruana, Rich (1998). "Multitask learning". In: *Learning to learn*. Springer, pp. 95–133.

Chen, Kai, Hang Song, Chen Change Loy, and Dahua Lin (2017). "Discover and Learn New Objects from Documentaries". In: *CVPR*.

Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille (2014). "Semantic image segmentation with deep convolutional nets and fully connected crfs". In: *arXiv preprint arXiv:1412.7062*.

— (2016a). "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *arXiv preprint arXiv:1606.00915*.

Chen, Liang-Chieh, Jonathan T Barron, George Papandreou, Kevin Murphy, and Alan L Yuille (2016b). "Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4545–4554.

Chen, Wenlin, James T Wilson, Stephen Tyree, Kilian Q Weinberger, and Yixin Chen (2015). "Compressing Neural Networks with the Hashing Trick." In: *ICML*, pp. 2285–2294.

Chen, Xinlei, Abhinav Shrivastava, and Abhinav Gupta (2014). "Enriching visual knowledge bases via object discovery and segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2027–2034.

Chen, Yunjin, Rene Ranftl, and Thomas Pock (2014). "Insights into analysis operator learning: From patch-based sparse models to higher order MRFs". In: *IEEE Transactions on Image Processing* 23.3, pp. 1060–1072.

Cheng, J., S. Liu, Y.-H. Tsai, W.-C. Hung, S. Gupta, J. Gu, J. Kautz, S. Wang, and M.-H. Yang (2017). "Learning to Segment Instances in Videos with Spatial Propagation Network". In: *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*.

Cinbis, Ramazan Gokberk, Jakob Verbeek, and Cordelia Schmid (2017). "Weakly supervised object localization with multi-fold multiple instance learning". In: *TPAMI* 39.1, pp. 189–203.

Dai, Jifeng, Kaiming He, and Jian Sun (2015). "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation". In: *ICCV*, pp. 1635–1643.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009a). "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009b). "ImageNet: A large-scale hierarchical image database". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 248–255.

Deselaers, Thomas, Bogdan Alexe, and Vittorio Ferrari (2010). "Localizing objects while learning their appearance". In: *ECCV*. Springer, pp. 452–466.

— (2012). "Weakly supervised localization and learning with generic knowledge". In: *IJCV* 100.3, pp. 275–293.

Deselaers, Thomas and Vittorio Ferrari (2010). "A Conditional Random Field for Multiple-instance Learning". In: *ICML*, pp. 287–294.

Doran, Gary and Soumya Ray (2014). "A Theoretical and Empirical Analysis of Support Vector Machine Methods for Multiple-instance Classification". In: *Machine Learning*, pp. 79–102.

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Evgeniou, Theodoros, Charles A Micchelli, and Massimiliano Pontil (2005). "Learning multiple tasks with kernel methods". In: *Journal of Machine Learning Research* 6.Apr, pp. 615–637.

Faktor, Alon and Michal Irani (2013). "Co-segmentation by composition". In: *ICCV*, pp. 1297–1304.

— (2014). "Video Segmentation by Non-Local Consensus voting." In: *BMVC*. Vol. 2. 5, p. 6.

Fei-Fei, Li, Rob Fergus, and Pietro Perona (2006). "One-shot learning of object categories". In: *IEEE transactions on pattern analysis and machine intelligence* 28.4, pp. 594–611.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine (2017a). "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *International Conference on Machine Learning (ICML)*.

— (2017b). "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *ICML*.

Fragkiadaki, Katerina, Pablo Arbeláez, Panna Felsen, and Jitendra Malik (2015). "Learning to segment moving objects in videos". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 4083–4090.

Franceschi, Luca, Michele Donini, Paolo Frasconi, and Massimiliano Pontil (2017a). "A Bridge Between Hyperparameter Optimization and Larning-to-learn". In: *NIPS 2017 Workshop on Meta-learning*.

— (2017b). "Forward and Reverse Gradient-Based Hyperparameter Optimization". In: *Proceedings of the 34th International Conference on International Conference on Machine Learning*.

Fu, Huazhu, Dong Xu, Bao Zhang, and Stephen Lin (2014). "Object-based multiple foreground video co-segmentation". In: *CVPR*, pp. 3166–3173.

Gidaris, Spyros and Nikos Komodakis (2018). "Dynamic Few-Shot Visual Learning without Forgetting". In: *CVPR*, pp. 4367–4375.

Girshick, Ross (2015). "Fast r-cnn". In: *ICCV*, pp. 1440–1448.

Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Glorot, Xavier and Yoshua Bengio (2010). “Understanding the difficulty of training deep feedforward neural networks”. In: *AISTATS*, pp. 249–256.

Gould, Stephen, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo (2016). “On differentiating parameterized argmin and argmax problems with application to bi-level optimization”. In: *arXiv preprint arXiv:1607.05447*.

Hajimirsadeghi, Hossein, Jinling Li, Greg Mori, Mohamed Zaki, and Tarek Sayed (2013). “Multiple Instance Learning by Discriminative Training of Markov Networks”. In: *UAI*, pp. 262–271.

Hariharan, Bharath and Ross B. Girshick (2016). “Low-shot visual object recognition”. In: *CoRR* abs/1606.02819.

Hariharan, Bharath, Pablo Arbeláez, Ross Girshick, and Jitendra Malik (2014). “Simultaneous detection and segmentation”. In: *European Conference on Computer Vision*. Springer, pp. 297–312.

— (2015). “Hypercolumns for object segmentation and fine-grained localization”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 447–456.

Hascoet, Laurent and Mauricio Araya-Polo (2006). “Enabling user-driven Checkpointing strategies in Reverse-mode Automatic Differentiation”. In: *arXiv preprint cs/0606042*.

Hazan, Elad et al. (2016). “Introduction to online convex optimization”. In: *Foundations and Trends® in Optimization* 2.3-4, pp. 157–325.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep residual learning for image recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick (2017). “Mask R-CNN”. In: *CoRR* abs/1703.06870.

Hochbaum, Dorit S and Vikas Singh (2009). “An efficient algorithm for Co-segmentation.” In: *ICCV*, pp. 269–276.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory”. In: *Neural computation* 9.8, pp. 1735–1780.

Hoffman, Judy, Deepak Pathak, Trevor Darrell, and Kate Saenko (2015). “Detector discovery in the wild: Joint multiple instance and representation learning”. In: *CVPR*, pp. 2883–2891.

Hong, Seunghoon, Hyeonwoo Noh, and Bohyung Han (2015). "Decoupled deep neural network for semi-supervised semantic segmentation". In: *Advances in Neural Information Processing Systems*, pp. 1495–1503.

Hong, Seunghoon, Junhyuk Oh, Honglak Lee, and Bohyung Han (2016). "Learning transferrable knowledge for semantic segmentation with deep convolutional neural network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3204–3212.

Horn, Roger A and Charles R Johnson (1990). *Matrix analysis*. Cambridge University Press.

Hsu, Kuang-Jui, Chung-Chi Tsai, Yen-Yu Lin, Xiaoning Qian, and Yung-Yu Chuang (2018). "Unsupervised CNN-based Co-Saliency Detection with Graphical Optimization". In: *ECCV*.

Huang, Jonathan, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. (2017). "Speed/accuracy trade-offs for modern convolutional object detectors". In: *CVPR*, pp. 7310–7311.

Huh, Minyoung, Pulkit Agrawal, and Alexei A Efros (2016). "What makes ImageNet good for transfer learning?" In: *arXiv preprint arXiv:1608.08614*.

Humayun, Ahmad, Fuxin Li, and James M. Rehg (2015). "The Middle Child Problem: Revisiting Parametric Min-cut and Seeds for Object Proposals". In: *Computer Vision (ICCV), IEEE International Conference on*. IEEE.

Ilse, Maximilian, Jakub Tomczak, and Max Welling (2018). "Attention-based Deep Multiple Instance Learning". In: *ICML*, pp. 2127–2136.

Jain, Suyog Dutt, Bo Xiong, and Kristen Grauman (2017). "Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos". In: *arXiv preprint arXiv:1701.05384*.

Ji, Shuiwang, Wei Xu, Ming Yang, and Kai Yu (2013). "3D convolutional neural networks for human action recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 35.1, pp. 221–231.

Jie, Zequn, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu (2017). "Deep Self-Taught Learning for Weakly Supervised Object Localization". In: *CVPR*, pp. 4294–4302.

Kaiser, Łukasz, Ofir Nachum, Aurko Roy, and Samy Bengio (2017). "Learning to Remember Rare Events". In: *arXiv preprint arXiv:1703.03129*.

Karpathy, Andrej, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei (2014). “Large-scale video classification with convolutional neural networks”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732.

Khoreva, A., R. Benenson, E. Ilg, T. Brox, and B. Schiele (2017). “Lucid Data Dreaming for Object Tracking”. In: *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*.

Kingma, Diederik P and Jimmy Ba (2015). “Adam: A method for stochastic optimization”. In: *International Conference on Learning Representations*.

Koch, Gregory (2015). “Siamese neural networks for one-shot image recognition”. PhD thesis. University of Toronto.

Koh, Yeong Jun and Chang-Su Kim (2017). “Primary Object Segmentation in Videos Based on Region Augmentation and Reduction”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 7417–7425.

Kolmogorov, Vladimir (2006). “Convergent tree-reweighted message passing for energy minimization”. In: *TPAMI* 28.10, pp. 1568–1583.

Krähenbühl, Philipp and Vladlen Koltun (2011). “Efficient inference in fully connected crfs with gaussian edge potentials”. In: *Advances in neural information processing systems*, pp. 109–117.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., pp. 1097–1105.

Lake, Brenden M, Ruslan Salakhutdinov, and Joshua B Tenenbaum (2015). “Human-level concept learning through probabilistic program induction”. In: *Science* 350.6266, pp. 1332–1338.

Larsen, Jan, Lars Kai Hansen, Claus Svarer, and M Ohlsson (1996). “Design and regularization of neural networks: the optimal use of a validation set”. In: *Neural Networks for Signal Processing [1996] VI. IEEE Signal Processing Society Workshop*. IEEE, pp. 62–71.

Le, T.-N., K.-T. Nguyen, M.-H. Nguyen-Phan, T.-V. Ton, T.-A. Nguyen (2), X.-S. Trinh, Q.-H. Dinh, V.-T. Nguyen, A.-D. Duong, A. Sugimoto, T. V. Nguyen, and M.-T. Tran (2017). “Instance Re-Identification Flow for Video Object Segmentation”. In: *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Li, Fei-Fei, Rob Fergus, and Pietro Perona (2006). "One-shot learning of object categories". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.4, pp. 594–611.

Li, Fuxin, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg (2013). "Video Segmentation by Tracking Many Figure-Ground Segments". In: *ICCV*.

Li, Ke and Jitendra Malik (2017). "Learning to Optimize Neural Nets". In: *arXiv preprint arXiv:1703.00441*.

Li, X., Y. Qi, Z. Wang, K. Chen, Z. Liu, J. Shi, P. Luo, C. Change Loy, and X. Tang (2017). "Video Object Segmentation with Re-identification". In: *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*.

Li, Yao, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel (2016a). "Image co-localization by mimicking a good detector's confidence score distribution". In: *ECCV*, pp. 19–34.

Li, Yi, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei (2016b). "Fully convolutional instance-aware semantic segmentation". In: *arXiv preprint arXiv:1611.07709*.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014a). "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer, pp. 740–755.

— (2014b). "Microsoft coco: Common objects in context". In: *ECCV*, pp. 740–755.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440.

Luketina, Jelena, Mathias Berglund, Klaus Greff, and Tapani Raiko (2016). "Scalable gradient-based tuning of continuous regularization hyperparameters". In: *International Conference on Machine Learning*, pp. 2952–2960.

Maclaurin, Dougal, David Duvenaud, and Ryan Adams (2015). "Gradient-based hyperparameter optimization through reversible learning". In: *International Conference on Machine Learning*, pp. 2113–2122.

Maninis, K.K., J. Pont-Tuset, P. Arbeláez, and L. Van Gool (2017). "Convolutional Oriented Boundaries: From Image Segmentation to High-Level Tasks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Märki, Nicolas, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung (2016). "Bilateral space video segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 743–751.

Maron, Oded and Tomás Lozano-Pérez (1998). "A framework for multiple-instance learning". In: *NIPS*, pp. 570–576.

Mishra, Nikhil, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel (2018). "A Simple Neural Attentive Meta-Learner". In: *ICLR*.

Mostajabi, Mohammadreza, Payman Yadollahpour, and Gregory Shakhnarovich (2015). "Feedforward semantic segmentation with zoom-out features". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3376–3385.

Munkhdalai, Tsendsuren, Xingdi Yuan, Soroush Mehri, and Adam Trischler (2018). "Rapid adaptation with conditionally shifted neurons". In: *ICML*, pp. 3661–3670.

Newswanger, A. and C. Xu (2017). "One-Shot Video Object Segmentation with Iterative Online Fine-Tuning". In: *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*.

Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han (2015). "Learning deconvolution network for semantic segmentation". In: *ICCV*, pp. 1520–1528.

Noh, Hyeonwoo, Paul Hongsuck Seo, and Bohyung Han (2016). "Image question answering using convolutional neural network with dynamic parameter prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 30–38.

Oord, Aaron van den, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. (2016). "Conditional image generation with pixelcnn decoders". In: *NIPS*, pp. 4790–4798.

Oreshkin, Boris N, Alexandre Lacoste, and Pau Rodriguez (2018). "TADAM: Task dependent adaptive metric for improved few-shot learning". In: *arXiv preprint arXiv:1805.10123*.

Papandreou, George, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille (2015). "Weakly- and semi-supervised learning of a DCNN for semantic image segmentation". In: *arXiv preprint arXiv:1502.02734*.

Papazoglou, A. and V. Ferrari (2013). "Fast Object Segmentation in Unconstrained Video". In: *ICCV*.

Pathak, Deepak, Philipp Krahenbuhl, and Trevor Darrell (2015). "Constrained convolutional neural networks for weakly supervised segmentation". In: *ICCV*, pp. 1796–1804.

Pathak, Deepak, Evan Shelhamer, Jonathan Long, and Trevor Darrell (2014). "Fully convolutional multi-class multiple instance learning". In: *arXiv preprint arXiv:1412.7144*.

— (2015). "Fully Convolutional Multi-Class Multiple Instance Learning". In: *ICLR Workshop*.

Pedregosa, Fabian (2016). "Hyperparameter optimization with approximate gradient". In: *International Conference on Machine Learning*, pp. 737–746.

Perazzi, F., J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung (2016). "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation". In: *Computer Vision and Pattern Recognition*.

Perazzi, Federico, Oliver Wang, Markus Gross, and Alexander Sorkine-Hornung (2015). "Fully Connected Object Proposals for Video Segmentation". In: *ICCV*, pp. 3227–3234.

Perazzi, Federico, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung (2017). "Learning video object segmentation from static images". In: CVPR.

Pinheiro, Pedro O and Ronan Collobert (2015). "Weakly supervised semantic segmentation with convolutional networks". In: *CVPR*. Vol. 2. 5. Citeseer, p. 6.

Pinheiro, Pedro O, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár (2016). "Learning to refine object segments". In: *European Conference on Computer Vision*. Springer, pp. 75–91.

Pont-Tuset, Jordi, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool (2017). "The 2017 DAVIS Challenge on Video Object Segmentation". In: *arXiv:1704.00675*.

Qiao, Siyuan, Chenxi Liu, Wei Shen, and Alan L. Yuille (2018). "Few-Shot Image Recognition by Predicting Parameters from Activations". In: *CVPR*.

Quan, Rong, Junwei Han, Dingwen Zhang, and Feiping Nie (2016). "Object co-segmentation via graph optimized-flexible manifold ranking". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 687–695.

Rahimi, Amir, Amirreza Shaban, Thalaiyasingam Ajanthan, Richard Hartley, and Byron Boots (2020). "In Defense of Graph Inference Algorithms for Weakly Supervised Object Localization". In: *arXiv preprint arXiv:2003.08375*.

Ramachandran, Prajit, Barret Zoph, and Quoc V. Le (2017). *Searching for Activation Functions*. Tech. rep. Google Brain.

Ranjan, Rajeev, Vishal M Patel, and Rama Chellappa (2017). "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ravi, Sachin and Hugo Larochelle (2017a). "Optimization as a model for few-shot learning". In: *International Conference on Learning Representations*.

— (2017b). "Optimization as a model for few-shot learning". In: *ICLR*.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*, pp. 91–99.

Rochan, Mrigank and Yang Wang (2015). "Weakly supervised localization of novel objects using appearance transfer". In: *CVPR*, pp. 4315–4324.

Roth, Stefan and Michael J Black (2005). "Fields of experts: A framework for learning image priors". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. IEEE, pp. 860–867.

Rother, Carsten, Tom Minka, Andrew Blake, and Vladimir Kolmogorov (2006). "Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs". In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 1. IEEE, pp. 993–1000.

Rudin, Walter (1964). *Principles of Mathematical Analysis*. Vol. 3. New York: McGraw-Hill.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *IJCV* 115.3, pp. 211–252.

Salakhutdinov, Ruslan, Joshua B Tenenbaum, and Antonio Torralba (2012). "One-Shot Learning with a Hierarchical Nonparametric Bayesian Model." In: *ICML Unsupervised and Transfer Learning*, pp. 195–206.

Santoro, Adam, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap (2016). "Meta-learning with memory-augmented neural networks". In: *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1842–1850.

Shaban, Amirreza, Alrik Firl, Ahmad Humayun, Jialin Yuan, Xinyao Wang, Peng Lei, Nikhil Dhanda, Byron Boots, James M Rehg, and Fuxin Li (2017a). "Multiple-instance video segmentation with sequence-specific object proposals". In: *CVPR Workshop*. Vol. 1.

Shaban, Amirreza, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots (2017b). "One-shot learning for semantic segmentation". In: *BMVC*.

Shaban, Amirreza, Amir Rahimi, Shray Bansal, Stephen Gould, Byron Boots, and Richard Hartley (2019a). "Learning to Find Common Objects Across Few Image Collections". In: *ICCV*, pp. 5117–5126.

Shaban, Amirreza, Ching-An Cheng, Nathan Hatch, and Byron Boots (2019b). "Truncated Back-propagation for Bilevel Optimization". In: *AISTATS*.

Sharir, G., E. Smolyansky, and I. Friedman (2017). "Video Object Segmentation using Tracked Object Proposals". In: *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*.

Shelhamer, Evan, Kate Rakelly, Judy Hoffman, and Trevor Darrell (2016). "Clockwork convnets for video semantic segmentation". In: *Computer Vision–ECCV 2016 Workshops*. Springer, pp. 852–868.

Shen, Yunhan, Rongrong Ji, Shengchuan Zhang, Wangmeng Zuo, Yan Wang, and F Huang (2018). "Generative adversarial learning towards fast weakly supervised detection". In: *CVPR*, pp. 5764–5773.

Shewchuk, Jonathan Richard (1994). *An introduction to the conjugate gradient method without the agonizing pain*.

Shi, Miaojing, Holger Caesar, and Vittorio Ferrari (2017). "Weakly Supervised Object Localization Using Things and Stuff Transfer". In: *ICCV*, pp. 3401–3410.

Simonyan, K. and A. Zisserman (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556.

Snell, Jake, Kevin Swersky, and Richard S Zemel (2017). "Prototypical Networks for Few-shot Learning". In: *Advances in Neural Information Processing Systems*.

Snoek, Jasper, Hugo Larochelle, and Ryan P Adams (2012). "Practical bayesian optimization of machine learning algorithms". In: *Advances in neural information processing systems*, pp. 2951–2959.

Srinivas, Niranjan, Andreas Krause, Sham M Kakade, and Matthias Seeger (2010). "Gaussian process optimization in the bandit setting: No regret and experimental design". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*.

Sung, Flood, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales (2018). "Learning to Compare: Relation Network for Few-Shot Learning". In: *CVPR*.

Tang, Kevin, Armand Joulin, Li-Jia Li, and Li Fei-Fei (2014). “Co-localization in real-world images”. In: *CVPR*, pp. 1464–1471.

Tsai, Yi-Hsuan, Ming-Hsuan Yang, and Michael J Black (2016). “Video segmentation via object flow”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3899–3908.

Uijlings, Jasper, Stefan Popov, and Vittorio Ferrari (2018). “Revisiting knowledge transfer for training object class detectors”. In: *CVPR*.

Vicente, Sara, Carsten Rother, and Vladimir Kolmogorov (2011). “Object cosegmentation”. In: *CVPR*, pp. 2217–2224.

Vinyals, Oriol, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. (2016). “Matching networks for one shot learning”. In: *Advances in Neural Information Processing Systems*, pp. 3630–3638.

Voigtlaender, Paul and Bastian Leibe (2017). “Online Adaptation of Convolutional Neural Networks for Video Object Segmentation”. In: *BMVC*.

Wang, Jane X, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick (2016). “Learning to reinforcement learn”. In: *arXiv preprint arXiv:1611.05763*.

Weiss, Yair and William T Freeman (2001). “On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs”. In: *IEEE Transactions on Information Theory* 47.2, pp. 736–744.

Wu, Zhengyang, Fuxin Li, Rahul Sukthankar, and James M Rehg (2015). “Robust video segment proposals with painless occlusion handling”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4194–4203.

Wulff, Jonas, Laura Sevilla-Lara, and Michael J Black (2017). “Optical Flow in Mostly Rigid Scenes”. In: *arXiv preprint arXiv:1705.01352*.

Xiao, Fanyi and Yong Jae Lee (2016). “Track and segment: An iterative unsupervised approach for video object proposals”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 933–942.

Xu, Ning, Brian Price, Scott Cohen, and Thomas Huang (2017). “Deep Image Matting”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yu, Fisher and Vladlen Koltun (2015). “Multi-scale context aggregation by dilated convolutions”. In: *arXiv preprint arXiv:1511.07122*.

Zagoruyko, Sergey and Nikos Komodakis (2016). “Wide Residual Networks”. In: *BMVC*, pp. 87.1–87.12.

Zhang, Dingwen, Junwei Han, Chao Li, and Jingdong Wang (2015). “Co-saliency detection via looking deep and wide”. In: *CVPR*, pp. 2994–3002.

Zhao, Hao (2017). “Some Promising Ideas about Multi-instance Video Segmentation”. In: *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*.

Zhuang, Bohan, Lingqiao Liu, Yao Li, Chunhua Shen, and Ian D Reid (2017). “Attend in groups: a weakly-supervised deep learning framework for learning from web data.” In: *CVPR*.